1992

The application of computer based instructions in MET

Soe Swe Aung
World Maritime University

WORLD MARITIME UNIVERSITY

Malmö, Sweden

THE APPLICATION OF COMPUTER BASED INSTRUCTIONS IN MET

By

AUNG SOE SWE

MYANMAR

A dissertation submitted to the World Maritime University in partial fulfilment of the requirements for the award of the

Degree of Master of Science
in
Maritime Education and Training(Nautical)

Year of Graduation

1992

I certify that all material in this dissertation which is not my own work has been identified and that no material is included for which a degree has been previously conferred upon me.

The contents of this dissertation reflect my personal views and are not necessarily endorsed by the University.

(Signature) ______________
(Date) 17-11-92

Supervised and assessed by:

Günther Zade

G. Zade
Professor
World Maritime University

Co-assessed by:

Samar J Singh

Capt. Samar Singh
Principle Lecturer
Department of Nautical Studies
Hong Kong Polytechnic

The Application Of Computer Based Instructions In MET

by

AUNG SOE SWE
MIYANMAR
MET (N) - '92

SPECIAL DEDICATION

TO

MY PARENTS, WIFE & SONS

AND TO

THOSE WHO LOVE TO TEACH IN

THE " MET " INSTITUTES

TABLE OF CONTENTS

INTRODUCTION AND SCOPE

CHAPTER I

WHAT IS CBI?

CHAPTER II

BASIC APPLICATIONS OF CBI

CHAPTER III

DEVELOPING AND DESIGNING LESSONS

CHAPTER IV

MICROCOMPUTERS AND MEDIA

PART ONE

PART TWO

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

LIST OF ILLUSTRATIONS

ABSTRACT

Today-computers had been developed from **Charles Babbage's Difference Engine** and **Herman Hollerith's Tabulating Machine,** which were invented many decades ago. The development nowadays is dynamic, both in hardware and software, it makes the prices fall dramatically. It became possible and reasonable for developing countries to use the computers in their MET institutions and start teaching the cadets and officers. I believe that, in one day, microcomputers may take the workload of a tutor in classes in different forms (e.g. full mission-simulators). Microcomputers may also become one of the solutions for the shortage of lecturers.

When the institution is furnished with computers, it is important to make sure that computers are used effectively. It is possible that teaching can be enhanced by using **CBI**(Computer Based Instructions) lessons and by using Computer Managed Learning**(CML)** systems.

There are advantages and disadvantages in CBI use. The teacher should know them how they can affect the students' learning psychology, so that he or she can avoid demotivation among the students. One should also be aware of *computerphobia* and should take care of the students who are *computerphobic*.

In designing the CBI lessons, it is important to have the knowledge of the use of text, color, and graphic for presenting information as well as formulating questions and providing feedback. In developing the CBI lessons, aiming at the achievement of creativity and student motivation, one should plan the procedure step by step to minimize the drawbacks.

There are eight steps needed to be taken for developing a CBI program. Like in a task analysis, the first step is to define the objectives of the subject that we are going to develop as a CBI program, after correctly selecting the level of the students. Details of the steps are explained in chapter III. It also explains the use of a hand scanner to produce the graphic and text that will be used in the lesson programs.

Following the step-by-step procedure, a tutorial program on "Lifeboat Equipments" has been produced. This is to share my knowledge to the other maritime educators as a practical experience.

Even the commercially produced software existing on the market is not perfect for every user nor are the CBI programs or this program. It is a start and is intended to encourage other lecturers in maritime institutions to begin to develop CBI program.

For further development of the teaching environment, we should look forward to using multi-media application packages that may include CD-ROM drive, sound card, video frame grabber card, etc.

Updating of knowledge is necessary and this can be done by:

(a) reading books, magazines on the computers and applications, and papers from conferences such as **IMLA**(International Maritime Lecturers' Association), **INSLC**(International Navigation Simulator Lecturers' Conference),

(b) participating in regional and international computer associations, and

(c) exchanging experiences among the institutes through writing and electronic mail.

ACKNOWLEDGEMENT

In preparation of this study, from beginning to the end, I believe that things had happened through miracles, from God. Since long ago, in many ways, he made me study this subject that I have never expected to work with it in this way.

I am sincerely thankful to Sasakawa Foundation for their financial support and constant care through out my study.

I am satisfied with my course professors Mr. J. Mulders and Mr. G. Zade for their guidance. Again, I am thankful to Prof. Zade as Academic Dean for his academic programs and field studies that were very valuable for my paper.

I am very pleased with the lectures on "BASIC programming" given by resident lecturer Capt. Wagner, the lecture on "Computers in MET" given by Capt. Samar Singh, and the lecture on "Administration of an MET Institution" given by Capt. Ferguson. They are supportive to my dissertation.

I have a sincere gratitude to my co-assessor Capt. Samar Singh for his close attention and guidance given to me while he was in Malmö and co-assessing my paper from Hong Kong, although he was busy. My former resident lecturer Capt. S. Cross is also to be thanked as he advised to do this tutorial program.

I acknowledge my colleague Mr. Parnupong from MET(E)- '92, Mr. Mohammad Shahba from MET(N)- '92, Capt. Soe Win from MSA(N)-'92, and Mr. Chingiz Teymurov from TMS- '93, for their assistance and participation in evaluating my tutorial instruction program on "Lifeboat Equipments".

I finally wish to thank the librarian, Mr. John Burne, his staff, the course secretaries and the university staff for their co-operation in helping me to complete my dissertation.

INTRODUCTION and SCOPE

THE POTENTIAL OF CBI IN DEVELOPING COUNTRIES

Shortage of nautical lecturers in the nautical institutes is becoming a common problem among our nations. This is sometimes critical, depending on the individual institute and situation. The problem seems bigger in the developing countries than in the developed world, because of different scenario between the two.

The scenario in developed countries and some fast growing developing countries can be described as follows.

(a) Shortage of marine officers, especially increasing dropouts from the training courses **(Glass, D, 1992)**.

(b) Lack of support from ship's owners for recruitment.

(c) Shrinkage of merchant shipping fleets.

(d) The marine officers move to other marine related jobs, not to the institutes.

The scenario in developing countries is as followed.

(a) Seafaring is a competitively popular profession to earn hard currency and the training institutes can not offer a satisfactory salary or facility to officers who wish to work ashore. This encourages them to go back to sea.

(b) Due to poor and unhappy working environment in the institutes, especially, government-run institutes, they go back to sea.

(c) The officers after saving enough money from sailing, change their profession completely to other kinds of business on shore, such as investing in construction works, owning restaurants, etc.

One should pay attention to the fact that shortage of officers means shortage of maritime lecturers.

The institutes are solving this problem in many ways. Among them are;

(a) persuading existing lecturers not to leave the institute by offering incentives(e.g. promotion, further studies, more vacations, etc.)

(b) binding them with bonds or contract and at the same time, recruiting new lecturers.

One of the remedies for this situation is to switch the classes to the computer lab and use the **CBI** *(Computer Based Instruction)* programs, provided the programmes have been developed beforehand.

We can not replace a teacher with CBI lessons permanently owing to its limitations. However, as a temporary remedy, a class or a combination of classes can be organized by introducing well-designed interactive CBI lessons.

DRAWBACK, LIMITATIONS AND ADVANTAGES OF CBI

While implementing the computer based instruction in the institute, we should first know the advantages and disadvantages of the CBI programs, and they can be outlined as followed.

The advantages

1. The immediate feedback provided by interactive terminals keeps students interested and eager to keep trying.
2. Weaker students are obliged to participate actively; they often remain passive in lectures.
3. The computer will wait patiently for an answer and will not express annoyance with a wrong response.

4. The graphics facility is a powerful aid in enhancing intuition, especially in giving insight into mathematical formulae.
5. Interactive graphics make it possible to sample many more illustrations than could easily be shown in a textbook.
6. Mathematical calculations can be done as readily for realistic examples as for artificially simple cases that can be solved analytically.
7. Large volumes of data can be handled with accuracy and without drudgery.
8. The novel techniques provide enrichment of a course through added variety.

The disadvantages

1. A programmer cannot cater for every possible response and may give unexpected and unhelpful responses to unusual input.
2. A few students are intimidated by the strangeness of a computer terminal (less likely for physics and chemistry students).
3. Packages can become boring if a student is alone at a terminal for too long; most packages should run for an hour or so.
4. A package will not be appreciated unless it has a perceived goal, and will not be considered important unless it is integrated into a course to the extent of being assessed by the teacher.

Garland and Munn (1983) point particularly to the advantages of animated graphics in the teaching of biology. Static diagrams, they claim, are often difficult to follow in this context. Modern computers have the capability to animate the graphics, which means they can be used, like a film, to show sequential changes and to build up understanding step by step. In addition, unlike a film, a computer program can be run at various speeds and is capable of revision and modification to suit individual needs.

WHY COMPUTER? WHY CBI?

Although there are disadvantages in the use of CBI, we should not disregard them, because computers became an integral part of our lives.

Nowadays, computers are a part of the university curriculum in every discipline. Students in the social sciences take quantitative and laboratory courses using computers instead of calculators and tachistoscope **(Rosen & Weil, 1990)**. Humanity students learn to write with word processors and produce artistic masterpieces with computer graphics. Languages are routinely taught through computer-based instruction programs as are many other topics.

Properly designed and implemented computer-based instructional materials that use gaming techniques-competition, scoring, time constraints, lively audio feedback, and colourful graphic can serve a dual purpose. Not only can such courseware provide students with a wealth of meaningful data, but it can also foster a degree of emotional involvement in the material to a degree that other presentation media can't.

So, why don't we, the maritime educators, use our computer resources that are at hand? Isn't it the time to change our monotonous teaching techniques to more interactive ways?

BRINGING CBI INTO YOUR HAND

If we consider that it is the time to start the application of CBI, we must first integrate the computer into our curricula..

PROCESS OF INTEGRATION

Assuming that computers are already installed in the computer lab, we can find the following interfaces. (see figure 1-1.)

1. Faculty - computer interface.

2. Student - computer interface.
3. Faculty - Student interface.

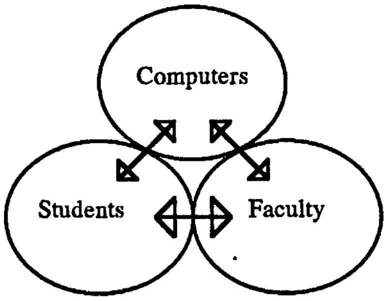


Figure 0-1. Three interfaces found in the universities and institutes.

PROBLEMS

The following problems can be found in our institutes and they should be solved first. The problems are as followed.

1. Faculty falling behind students in term of computer literacy.

2. Faculties, in contrast to students, are learning to use computers to do different things for different purposes and are tending to become more narrow and specialized in their computer literacy. For example, there are faculty who uses microcomputers to do advanced statistical analysis of research data, but cannot copy one disk to another.

3. Students seem to be learning a wider variety of computing skills and uses, but do not achieve much depth in any one area of computing **(Hofman, 1991)**.

Figure 0-2 shows what it should be in reality.

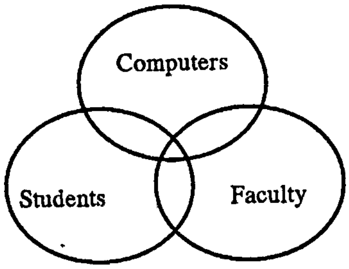


Figure 0-2. What should be happening.

Figure 0-3 shows what is happening in our institutes and universities.

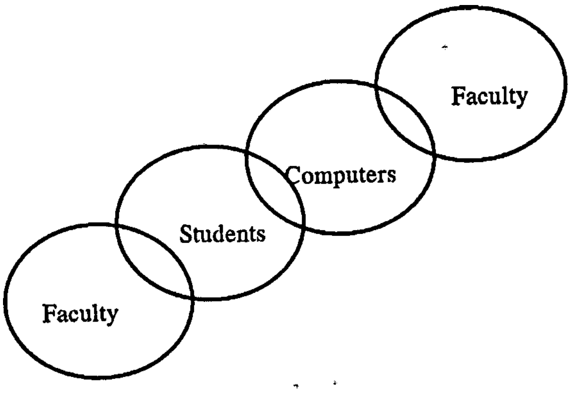


Figure 0-3. What is happening.

Faculties are interacting with computers, and faculties are interacting with students, but not both at the same time nor in the same context. All too often the faculty - student interaction is not meaningful or helpful to the student-computer interaction. In many cases, it confounds it.

ANOTHER PROBLEM: COMPUTER-UNFRIENDLINESS

In light of friendliness with computer, we should know that there are some students who have an aversion to computer and anxiety over using them. This tendency is called *computerphobia*.

Jay(1985) defined computerphobia as:

(a) a resistance to talking about computers or even thinking about computers,

(b) fear or anxiety toward computers, and

(c) hostile or aggressive thoughts about computers.

It is not easy to give an answer to a question "who is the computerphobic?" It has been found by researchers that it centres on several key demographic features(e.g. Gender, age, ethnic background); personality factors(e.g. Locus of control, mathematics anxiety, state and trait anxiety, and computer experience.

Larry D. Rosen and Michelle M. Weil(Department of psychology, California State University) have recently investigated the level of computerphobia among 585 elementary and secondary teachers in six Southern California school districts. They did an in-depth interview study of 40 computerphobics and 20 computer users to examine the aetiology of computerphobia. They mentioned that the etiologic roots of computerphobia were found as early as childhood in initial reactions to mechanical and computer experiences.

They believe that the results of their study will lead to future intervention programs for the computerphobic teachers. So, maritime educators should bear in mind that the problem of computerphobia among teachers and students is common, and I believe we can solve this problem by implementing creative CBI programs and encouraging those computerphobics to change their attitudes toward the computer.

THE SCOPE

I assume that all readers of my dissertation have some familiarity with computers; in short, that they are *"computer literate"*. You should have some idea of what computers are generally designed to do; that is, accept input, process data, and output result. You may already have a feel for what types of operations computers can be easily programmed to perform and what types are more difficult to implement.

Knowledge of a computer programming language will be helpful, but more from a conceptual point of view than a technical one. My work contains very few programming examples, and these are presented algorithmically so that they can be understood by anyone who has had minimal exposure to computers.

I also assume that readers have some knowledge of educational technology, the "soft" science- of communicating information from one person to another.

The objectives of my dissertation are as followed.

1. The readers will learn the main elements of CBI, required for effective learning. (See chapter I).
2. The readers will learn how to design the screen for CBI programs. (See Chapter I).
3. The readers can make a decision on which type of application(e.g. Tutorial, Drill, Simulation) he should choose to suit the student's need within the resources that he has. (See chapter II).
4. The readers will have a clear knowledge of Computer Managed Learning(**CML**). (See Chapter II)
5. The readers can design and develop their CBI lessons on maritime subjects systematically either by
   a) . software, or
   b) . programming language, or
   c) . authoring language, or
   d) . a combination of any of the choices mentioned above. (See Chapter III).
6. The readers will acquire a feel for the development of computers in the maritime field and the future of CBI as a medium(Chapter IV).
7. In Chapter V, the conclusion, maritime educators are strongly advised to make use of the computer in MET, after being informed of its limitations and advantages, and how to develop CBI programs, in the previous chapters. The assistance and up dating the CBI programs can be done through international and regional associations. (See Chapter V)

THE DEVELOPMENT OF CBI IN MYANMAR INSTITUTE OF MARINE TECHNOLOGY

The Institute of Marine Technology, **IMT** in short, is located in the capital city of Yangon(long known as Rangoon), on the river bank of Yangon river.

It is established with such powers and functions so as to educate students to become certificated officers of international standard in accordance with the International Maritime Organization conventions.

The training courses at the institute are designed not only to provide training in the relevant maritime field but also to give emphasis on discipline and formation of responsible mariners to meet the requirements of the national shipping line as well as the international shipping lines.

The faculty has three departments, namely; **The department of nautical science, the department of marine engineering,** and **electronic department.**

The nautical department provides profound theory and practice for the safe and economic operation of ships. It designs courses that provide a thorough education in the subjects of specialist studies.

The courses under the responsibility of this department are ranging from cadet training, and rating training to master (foreign-going) refresher courses.

The courses conducted by the engineering department are ranging from Basic engine room rating course and engineering cadet course to first class steam and motor engineering classes.

The electronic department jointly conducts courses with the Nautical department, such as Radar observer course, Radar and ARPA simulator course and Electronic navigational Aid course.

SIMULATORS IN NAUTICAL DEPT.

Radar and ARPA-Simulator

The recently installed Radar simulator has two ownships expandable to four. An ARPA is attached to one ownship. The radars fitted in the ownship modules are FURUNO and the software is run on the Hewlett Packard computer-9000 series. There is a colour video display in the instructor's console which can be switched over to any ownship anytime. A colour plotter is connected to the computer to plot the tracks of the vessels in the scenario on the paper for evaluation.

Electronic Navigation Aids-Simulator

There is a simulator for electronic navigation aids which is installed in the same year(i.e. 1989). This includes an *Echo-sounder*, an *Automatic Direction-finder*, a *Loran* receiver, an *Omega* receiver, a *satellite navigator*, a *Decca* receiver, and two video plotters. The software is run on a *Hewlett Packard* computer-9000 series which is separate from that used in radar simulator.

Loadmaster

There are two loadmaster-computers supplied with a software for four different types of ships, namely;

(a) Container ship model,

(b) Bulk carrier module,

(c) Tanker model, and

(d) General cargo model.

Software is run with Concurrent-Dos, and modification of existing models or new ship model can be done by **FORTRAN**[1] programming language.

COMPUTERS

At the end of 1991, there are five computers in our institutes. Three of them are NEC-brand and the other two being

IBM-compatibles imported from Singapore. Both have a hard disk of 80 mega bytes, and two floppy disks with colour monitors.

In the coming year, the computer room will be integrated to a computer-lab and a Local Area Network(**LAN**) will be installed, due to the increasing number of computer classes.

So far, the administration of the institute is semi-computerized. The staff have to come to work in the computer room if they have something to do with the computer, e.g. Accounting, word-processing, database and spreadsheet calculation, etc. In the near future, the administration office will be installed with computers connected to the **LAN**.

The budgets for the development of computers and part of the maintenance costs are granted by the government and a small portion of money is derived from tuition fees.

Computer training courses are given by part-time lecturers from the Universities Computer Centre(**UCC**) in Yangon. Since 1991, computer courses are integrated in the curricula of the cadet's training and officer's refresher training courses. There is a short computer-user's course that is a one-month course for people(outsiders) not attending any training courses in the institute.

The softwares existing in the institute is listed as followed.

Word processor		
Name	*Version*	*Where used*
WordPerfect	5.1	all courses
Word for windows	2.0	future courses

Database		
Name	*Version*	*Where used*
Dbase 3 plus	1	all courses
Dbase 4		all courses

Spreadsheet		
Name	*Version*	*Where used*
Lotus 123	1.3	all courses
Lotus for windows		future courses

Drawing tool		
Name	*Version*	*Where used*
Harvard Graphic	1	Senior, outsiders
Autocad	10	Outsiders

Programming		
Name	*Version*	*Where used*
GW-BASIC		all courses
Quick Basic	4.5	future courses
Language-C		future outsiders

Accounting		
Name	*Version*	*Where used*
Dac-Easy		outsiders, staffs

Proposed softwares and hardware that are to be purchased for application of CBI, are as followed.

Name	*Type of software & hardware*	*Where to use*
PC Storyboard-Live	Multi-media presentation, animation,	all courses
Video card, Sound card	Multi-media	all courses
CD-ROM drives & discs	Multi-media	

CBI programs are not yet developed in my institute. As the prices of the computers, their hardware and software are falling tremendously, our institutions have a better opportunity to purchase hardware and software according to our needs.

Firstly, we have to convince our budgeting authorities or financing industries to support our implementation program of the CBI lessons by showing them the effectiveness of the programs.

Secondly, as a next stép, we should plan to integrate our computer resources to **multi-media** learning programs.

[1] **FOTRAN**:-Formula Translator Language. (It is a high level language).

CHAPTER I

WHAT IS CBI?

A BRIEF HISTORY OF COMPUTERS IN EDUCATION

The first use of computers by educational institutions was in the 1950s. The computers were used for administrative purposes as well as for instruction research at that time. One famous such research was called **PLATO(Program Logic For Automated Teaching Operations)** project that began in 1960 with the goal of designing a large computer-based system for instruction.

Later, IBM introduced *Coursewriter*, a programming language designed for preparing instructional materials on IBM's large computers.

With the falling price and numerous production, more schools and colleges began using computers in the 1970s for administration functions. The instructional use was found only in universities, military and a few private companies.

In 1972, the **MITRE** Corporation and Brigham Young University started development of the *Time-shared Interactive Computer-controlled Instructional Television System* **(TICCIT)**. With TICCIT, students study lessons presented on standard color televisions and interact through modified typewriter keyboards, all of which are controlled by a minicomputer. The TICCIT which is intended for adult instruction embodied an instructional philosophy called *Learner-controlled Instruction (LCI)*. The TICCIT system continues to be used in a variety of adult educational environments, mainly in the military.

During this period the PLATO project introduced **PLATO IV**, a large time-shared instruction system. Students study on individual terminals, hundreds of which are connected to a large computer on which all lessons and student data are stored. All program execution occurs on the main computer, which communicates with the terminals on the telephone line. PLATO IV allows up to 600 students to use the computer simultaneously. In addition, it allows authors to develop instructional materials at the same time the students are studying lessons. **LOGO**[1] was one of the other famous CBI projects developed in the early seventies.

Many of these computer projects grew rapidly and both PLATO system and TICCIT became commercially available during this period.

In the mid-1970s a few small computer companies began experimenting with microcomputers. In 1977, three microcomputers produced by **Radio Shack, Commodore Business Machines** and a comparatively unknown company called **Apple Computer**, were proved to be successful. All three were well marketed, the Apple Computer being best.

From 1977 to now, we have seen phenomenal growth in the educational uses of computers. Most colleges and universities have microcomputers and most Nautical institutes have them, too.

As the computer revolution continues, both education and training are being increasingly influenced by the presence of readily available computing resources **(Alessi & Trollip 1985)**.

[1] **LOGO**, which was developed by **SEYMOUR PAPERT** at the Massachusetts Institute of Technology, was powerful but easily learned programming language for children.

THE APPLICATION OF CBI

Nowadays, schools and colleges are buying small computers at a remarkable rate. Their primary use generally is to:

(1) make teachers and students computer literate,

(2) teach them the fundamentals of how computers work, and

(3) to introduce them to programming.

In industries and other institutions that are concerned with training, their focus has been more on *teaching with computers*. Therefore, in our nautical institutes the focus should be on teaching with computer, and that is **CBI**. If it is the case where the students need to be computer literate, as a prerequisite for the CBI, they should be given a short computer-literacy course before going with the CBI.

If necessary, depending on the students' need and the facilities available, *job analysis* should be performed before any decision is made to set up a computer course. Assistance and advice should be sought from the University of Computer Science or similar institutions in the country.

TEACHING ABOUT COMPUTERS

Nowadays, in most of the nautical institutes, the use of computers may be limited to word processing, databank and accounting. When computers are available for teaching and learning in the institute, we should realize that there are two parts in "Teaching about computers" namely;

(1) Computer literacy for non-professionals and

(2) Computer Science.

Computer literacy is a wide subject and therefore its scope will vary with the time available to study. The main objective of the computer literacy is to make people comfortable with computers. Students should learn how to operate them, should know their applications, and should understand about computer ethics and the use and misuse of computers. But here in programming, some educators argue that it is a disservice not to teach programming because that is where the job market is. Others respond that teaching programming can wait until the student makes a choice to pursue a career in a computer field.

TEACHING WITH COMPUTER

Apart from CBI there are a large variety of acronyms, such as:

CAI :	Computer-aided Instruction
CAL :	Computer-Aided Learning
CBT :	Computer-based Training
CML :	Computer-managed Learning

CBI and CAI are seen to be as instructions given to computers for interactive teaching with students while CAL lies more on the student's side or strategy.

Barker(1989) described as *"the phrase Computer assisted Learning (CAL) has become to be regarded as something of an umbrella term that is used to describe, collectively, the many different teaching, training and learning applications of computers."*

Dean and Whitlock(1989) mentioned that, CBT is derived from two specialist areas - educational technology and computing and that CBT offers many advantages over traditional training methods and provides an increasingly popular solution to a wide range of training needs.

Beard and Hartly(1984) mentioned that the terms were too many to list and that the familiar definitions of **computer-assisted learning(CAL)** and **computer-managed learning(CML)** were therefore extended, using the headings prepared by a committee for a conference of the National Association of Users of Computer Aids to Learning in 1977. By then, CML was translated into *"learning support systems"*.

IMPORTANT ASPECTS OF CBI

CAL includes both *learning through* the computer (such as drill and practice, diagnostic testing, tutorial, etc.) and *learning with* the computer (such as simulation and games, problem-solving, creative activities, etc.)

CBI categorizes its type into four; namely,

(a) **Tutorial Instructions,**

(b) **Drills,**

(c) **Simulations,** and

(d) **Tests**

However, when developing any of above-mentioned CBI programs the following four phases must be present, namely;

(a) Presentation of information to student,

(b) Guiding the student,

(c) Allowing the student to practice and

(d) Assessing student's learning of CBI.

Any of the above-mentioned phases should never be neglected whenever developing an effective Computer based instruction. One may worry and ask questions as how should the students be guided? And how can response from students be assessed?

The fundamental factors to be considered are described as followed: **(Alessi & Trollip, 1985. Pg. 60 -62)**

PRESENTING INFORMATION

(1) Depending on the type of information to be presented, the forms of presentation are different. For verbal or pictorial information, an instructor may present rules and principles, show pictures, or provide other non-verbal information. To teach such skills as taking a bearing by an *azimuth-mirror*, the instructor will model the skills to be learned.

(2) An important part of presenting information is through example. Thus, when teaching the *stopping distance* of a ship, the instructor should demonstrate the following fact. "The momentum of a ship when loaded is more than that when she is in ballast-condition".

(3) In some cases, teaching does not begin with the presentation of information. Some models are based on the assumption that students should discover principles or develop skills through experimentation and practice. Based on past research, models that begin with the presentation of information are more efficient and demonstrably more successful **(Koran, 1971; Merrill, 1974; Klausmeier, 1975).**

GUIDING THE STUDENT

(1) In the classroom, guidance often takes the form of the instructor asking questions that students must answer. When the answer is wrong, the instructor may either tell the student the correct answer or may ask leading questions to help the student recall the correct information.

(2) When the student learns from the book, questions are sometimes included as guidance. However, in CBI, unlike the classroom, if the student does not perform correctly, true guidance does not occur. The student may receive help only at a later stage, when the instructor looks at what the student has done and provides feedback.

(3) Guidance is important in CBI because no student learns all that is taught on a single exposure. Students will make errors and frequently be unaware that they have made them. It is necessary that the students be made aware of these and correct them. Therefore guiding the student is very interactive. In this process, the student is attempting to apply new knowledge. The instructor is correcting and guiding and the student is making further attempts. It is often omitted in the instructions and yet

is the most important part of the instruction.

PRACTICE

(1) The student needs to learn information permanently rather than for a short time. Practising a skill a single time or answering a single question will not guarantee retention.

(2) Fluency and speed are related but they have slight different aspects on well-learned information. Some information does not require speed or fluency. However, the student should remember how to do things. Therefore students can gain retention from practice.

(3) Many activities of practice can be found in classroom instruction. Workbooks are the most common method of practising in classroom. When students make an error practising with a workbook, the error might never be corrected. In foreign-language instruction, a common type of practice is the flash-card drill. The student produces a pile of cards, for example, with French words on one side and the equivalent English words on the reverse side. The student then goes through the deck of cards, trying to translate the words correctly, and receives immediately corrective feedback by looking at the other side of the card.

ASSESSING STUDENT LEARNING

(1) The above-mentioned phases are considered as instruction. However, we should not assume that instruction will be successful for all students. Student learning should be assessed usually by tests, which are an important part of the instructional process.

(2) There are two main purposes of assessment; to aid the learners in their subsequent learning, and to

report on what they have already learned- e.g. as a grade or as a written report on each learner**(Rowntree 1990. 302)**. Tests provide the quality of teaching, as well**(Alessi & Trollip 1985. 62)**.

(3) Assessment is used in education in different ways depending on how the information is delivered to the learner. *Formative assessment* and *summative assessment* are used in open-learning**(Rowntree 1990)**. In higher education today, there is more emphasis on evaluating the process of learning, and less on what is recalled or retained at the end of a course, and a greater interest in the contributions that can be made with continuous or course-work assessment. Three types of assessment can be found in higher education, namely; *continuous assessment, coursework assessment and self-and peer assessment***(Beard & Heartly 1984. 273)**.

The *formative assessment* has two aspects, namely; **(a)** It is used when learning about a learner to decide what to say next to that learner. Thus, the learners can also assess their strengths and weaknesses to decide where next to put their efforts. This assessment may involve *peer assessment* in which students are involved in assessing their colleague's work(e.g. marking their colleagues' papers) after which they assess their own work according to the marking schedule which is prepared in advance by the staff. **(b)** Sometimes the learner may be required to tackle occasional tests and examinations and these will be assessed by a tutor. Then the tutor may provide comments that may help the learner to learn. One of the functions of these more formal exercises may be to let the learners measure their competence by the standard expected by some competent body.

The *summative assessment* sums up what each student has achieved. The purpose of this assessment is to report, or put on records, what is known of the learners' attainments. What did they study? Did they pass or fail?

What grades did they get? What competencies did they acquire? Are they now fit to work without supervision? Here the prime purpose is to establish what the learners have achieved - not to give them feedback that will help them learn **(Rowntree 1990)**.

As a summary, there are many options that the computer may serve any combination of the four phases. It may present initial information, after which the student receives guidance from an instructor and practices using a workbook.

The student may learn initial information from a lecture, after which the computer is used to practice some parts of the material to fluency.

The computer may be used for the first three phases, with testing being done in the traditional way by the instructor. In all cases, the four parts of instruction should be present, though they may be embodied in any combination of different media. Especially, when the computer is responsible for the total instruction, it is important that all four phases be included.

DESIGNING THE SCREEN FOR CBI

As CBI use monitor or TV as display unit for presenting information, the proper and systematic layout of the screen is important.

The display should contain a number of standard components, whether the instructional sequence involves drill and practice, simulation, problem solving or, simply presentation of new information. Such components include:

orientation information,
directions,
student responses,
error messages, and
student options **(Heines 1984)**.

ORIENTATION INFORMATION

The form or layout of the screen regarding to information must be oriented in such a way that students can usually figure out where they are. They can lose their awareness completely especially when the program branches out.

The *orientation information* can be supplied by putting the current module name and lesson name at the top of the screen

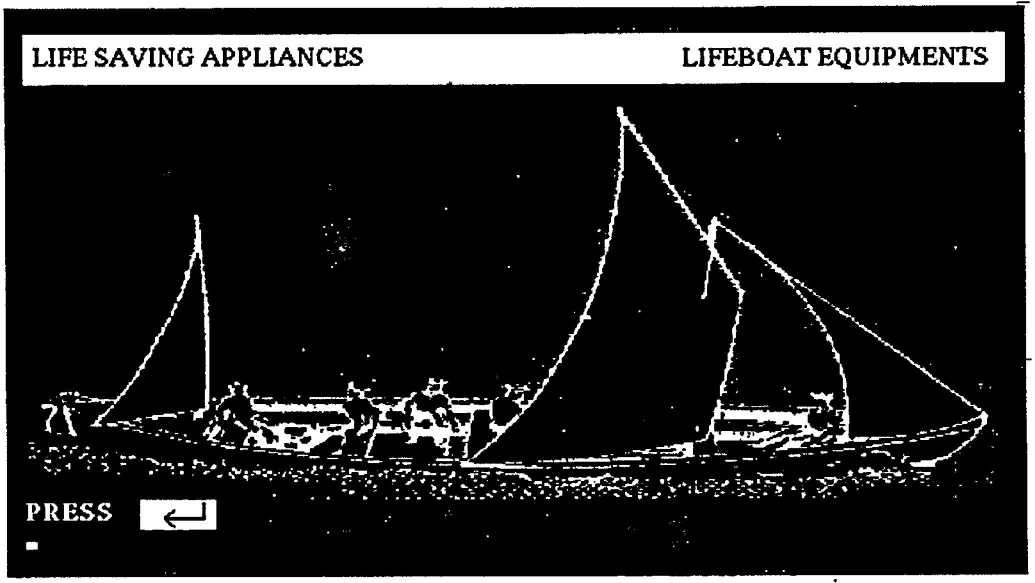


Figure 1-1. Orientation Information in Reverse Video at top of the screen.

Normally, they can be displayed as small text, quiet color or reverse video (dark letters on a white background)(see figure 1-1). In a large scale computer based training, besides module and lesson names, frame numbers are also displayed. These are useful only to the program development team as a reference back to the relevant screen design sheet. For the students, it can be misleading and distracting**(Dean & Whitlock, 1989)**.

STUDENT DIRECTIONS AND RESPONSES

Directions are the instructions the student has to do and responses mean anything the student types. These two components are considered together because they are often adjacent or share the same functional area. The instruction ***"PRESS RETURN"*** (or **PRESS ↵**) is the simplest and most commonly used for instruction to go forward. In some CBI programs, it can be found in the *prompt area*(see figure 1-2). Both of these components are best reserved for a specific functional area. By so doing, students always know where to look for directions, even when the screen is changing rapidly and where to expect their typed responses to appear.

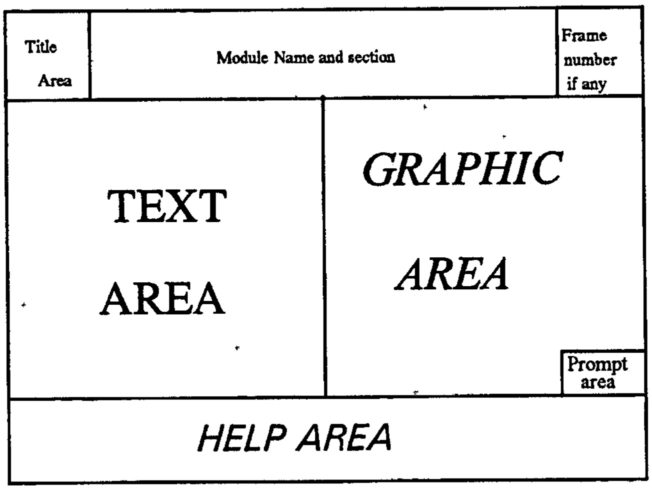


Figure 1-2. Typical functional areas on a screen.

ERROR MESSAGES

The error messages referred to here are not feedback that explains incorrect responses, rather, they are messages printed in response to inappropriate student entries that the computer cannot accept as answers.

The construction of error messages is an art in itself. These messages should be meaningful and short. Some used two levels of error massages. When an error is made, the first level is displayed.

This message is limited to a few key words(not more than one line) on the screen. The student can take one of two actions; either re-enter his or her response or type a special character for more detailed explanation of the error. A specially designated **HELP** key can be provided to access to the second level(like question mark).

STUDENT OPTIONS

There are many options that the students want to do. They are: (a) to exit the program, (b) request help, (c) enter a comment for their instructor, (d) view a summary of the material presented, (e) go back and review a specific section, etc.**(Heines 1984)**.

In some CBI programs, these options are put in the help area that is one of the functional areas(see figure 1-2). The novel way of displaying student options is by a menu.

As a summary, one should consider the amount of computer literacy to be given to the students and pay attention to the enhancement of screen design to increase their learning. Next chapter will illustrate the reader on different types of CBI applications and their strategies.

CHAPTER II.

BASIC APPLICATIONS OF CBI

There are four ways of teaching by CBI as mentioned above. Basic rules and principles for each application are explained here.

TUTORIAL

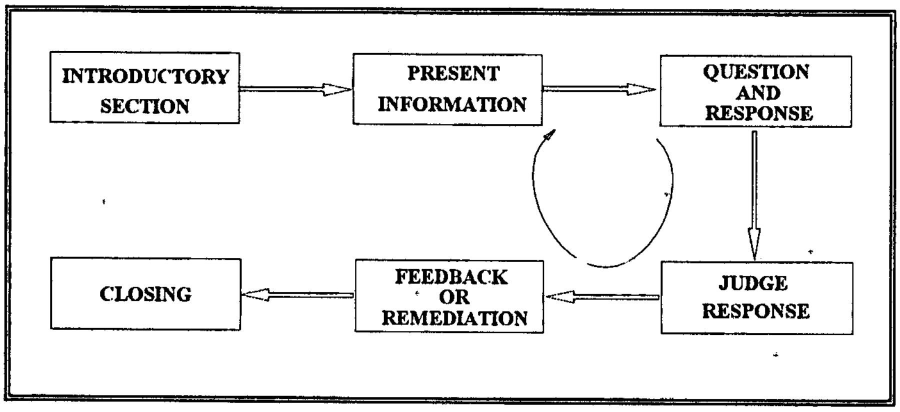


Figure 2-1. Tutorial.

Tutorial Instructions are programs that teach by carrying on a dialogue with the student.

FEATURES: -

- They present information. They ask the student questions and make decisions based on the student's comprehension whether to move onto new information or to engage in review and remediation.

USE: -

Tutorials are used in almost every subject area from the humanities to the social and physical sciences. They are appropriate for presenting factual information, for learning rules and principles, or for learning problem-solving strategies **(Gagne, Wager, & Rojas 1981)**.

SEQUENCES IN BRIEF: --

Figure 2-1 shows the structure and sequence of a tutorial. It begins with an introductory section that informs the student of the purpose and nature of the lesson. After that a cycle begins. Information is presented and elaborated. A question is asked that the student must answer. The program judges the response to assess student comprehension and future performance. At the end of each iteration, the program makes a sequencing decision to determine what information should be treated during the next iteration.

The cycle continues until the lesson is terminated by either the student or the program. At that point, which we call the closing, there may be a summary and closing remarks.

Generally, to produce an effective Tutorial program, **Alessi & Trollip. (1985)** pointed out that the following characteristics should be included.

- *Introduction*
- *Presenting* of information

- *Question and responses*
- *Judging* responses
- Providing *feedback* about responses
- *Remediation*
- *Sequencing* lesson segments
- *Closing.*

INTRODUCTION OF A TUTORIAL

The followings should be included in the introduction of a tutorial.

- ♦ The title page
- ♦ Presentation of objects
- ♦ Directions
- ♦ Stimulating prior knowledge
- ♦ Initial student control of sequence, and
- ♦ Pretesting

TITLE PAGE

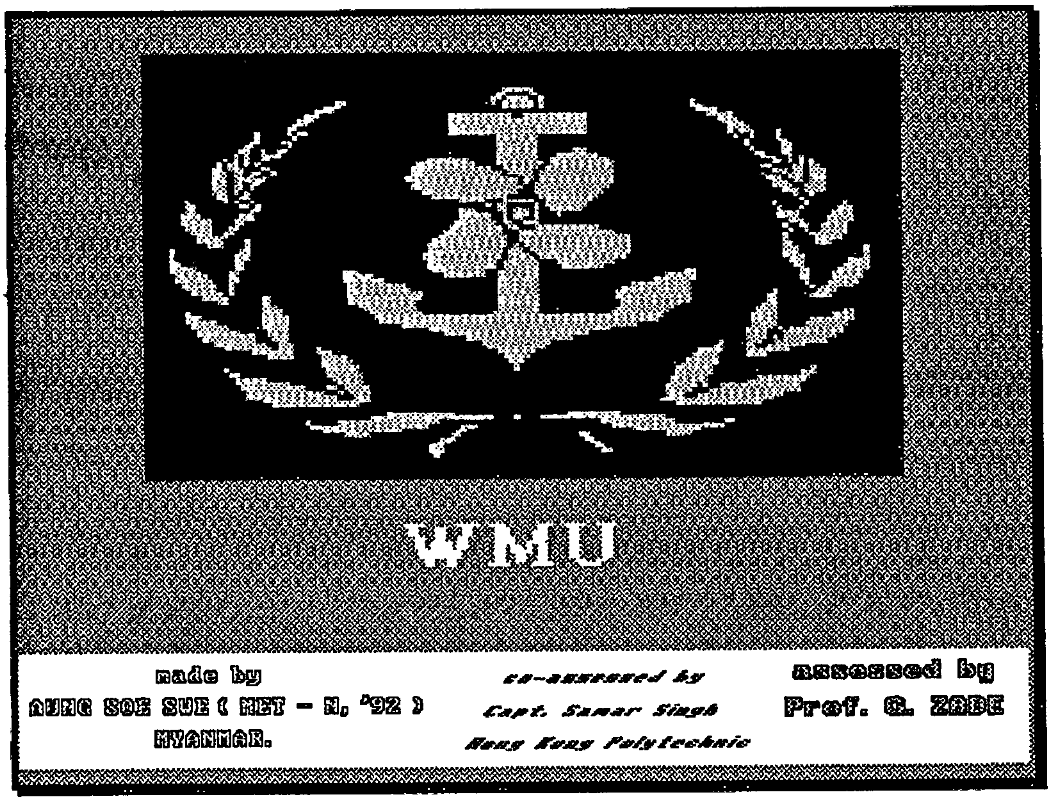


Figure 2-2. Title Page.

The purpose of putting a title page in a tutorial is (a) to attract the student's attention, (b) to make interesting and (c) to indicate briefly what the lesson is about.

Title pages can be a simple one with a title and the author's name or can be with colors and/or animated graphics. However, long, complicated or humorous text and graphics can be distracted and should be avoided. It should be noted that humorous and flashy title pages may be greeted only at the first time encountered by the student, not subsequent pages.

PRESENTATION OF OBJECTIVES

The objectives should be in behavioral form. The statement of the lesson's objectives should follow frequently the title page.

Use action words such as *choose, abandon, use*, etc. However, sometimes objectives are omitted due to oversight. When objectives are not given to the student, it is still useful to have them in a printed document for the instructor.

DIRECTIONS

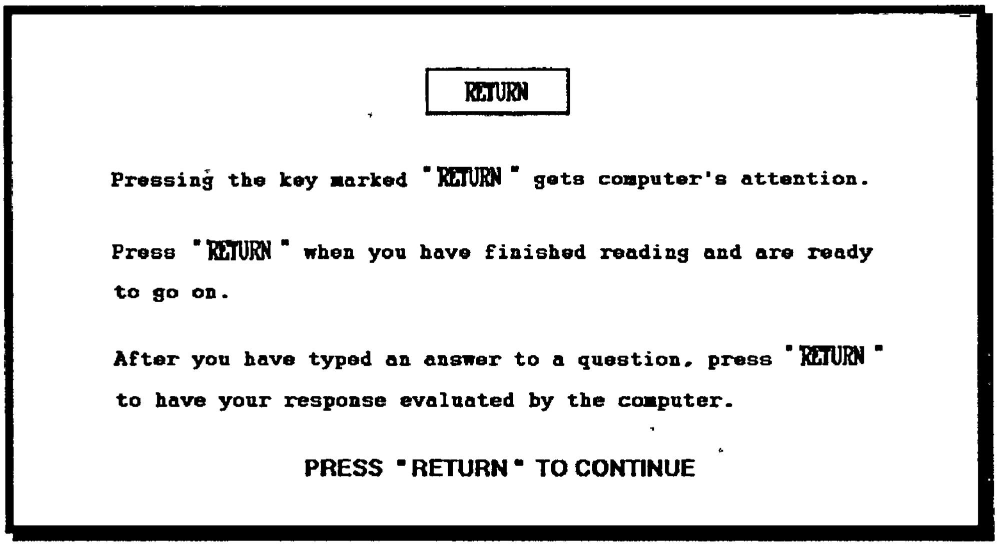
Directions should be concise and accurate. They should include information about :-

(a) how to move forward,

(b) how to back up if allowed,

(c) how to answer questions, and

(d) how to get help.



RETURN

Pressing the key marked "RETURN" gets computer's attention.

Press "RETURN" when you have finished reading and are ready to go on.

After you have typed an answer to a question, press "RETURN" to have your response evaluated by the computer.

PRESS "RETURN" TO CONTINUE

Figure 2-3. Directions for moving forward.

The directions should contain a way to return to the directions from any point in the lesson. It is of utmost important especially for students for the first time.

STIMULATING PRIOR KNOWLEDGE

Students will learn more if they can relate new information to what they already know **(R.C. Anderson 1977; Adams & Bruce 1980; Rumelhart & Ortony 1977)**. A tutorial should not review prior knowledge in detail, but should provide a synopsis of material studied previously.

INITIAL STUDENT CONTROL

The most common way of giving the student this choice is with a menu**(Alessi & Trollip 1985)**. In Figure 2-4, pressing [0] allows the student to go out to DOS prompt, and pressing [2] sends the student to the section explaining how the lifeboat equipments are fixed and used. Depending upon the complexity of the subject matter and the student's ability to control the instructions, The initial choices can be either simple or complicated. However, at least three choices should be included, namely; (a) the choice to return to the menu after each section is completed, (b) the choice to return to the directions(if the lesson is rather long) at any time, and (c) the choice to end the lesson (closing down).

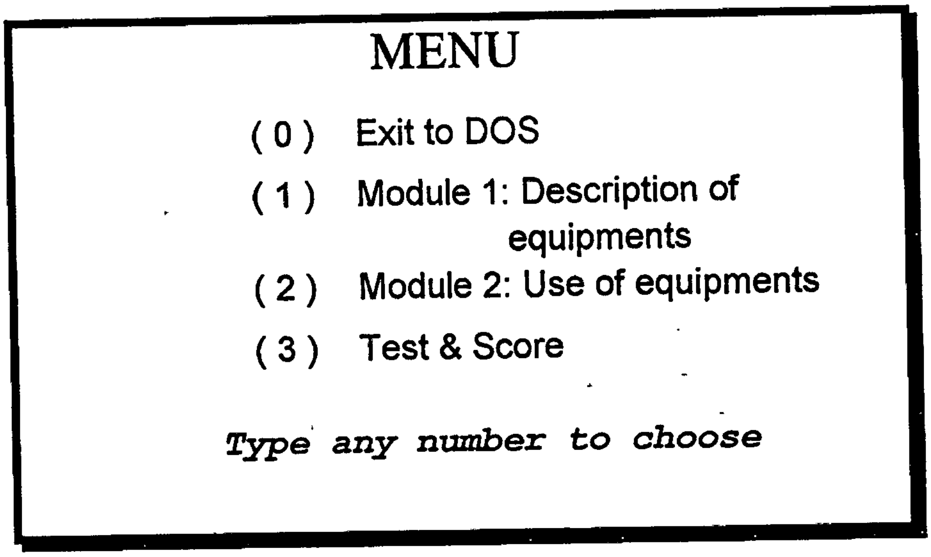

MENU
(0) Exit to DOS
(1) Module 1: Description of equipments
(2) Module 2: Use of equipments
(3) Test & Score
Type any number to choose

Figure 2-4. Initial Menu for a Tutorial.

PRETESTING

The purpose of a pretest is to make sure whether the student is ready to and should study this particular lesson or should skip this lesson. Pretest is not necessarily be included in the lesson, but it can be in a separate program. There are reasons for having the pretest separate. They are as followed - (a) Some. teachers do not like having pretest. (b) The pretest will have many questions which the students have not studied and they will do poorly; this may damage the students' attitudes or make them apprehensive about the difficulty of the new subject. Therefore it is recommended to keep the pretest separate with the lesson.

PRESENTATION OF INFORMATION

Presentation plays a big role to motivate the student. Mode of presentation involves text, graphics and sound. Text is the most common and easiest way of presentation. Graphic presentations include any visual information that is not text, such as pictures, line drawings, cartoons, bar graphs, line graphs, photographs and animated images. The use of sound in CBI is not as advanced as the visual mode.

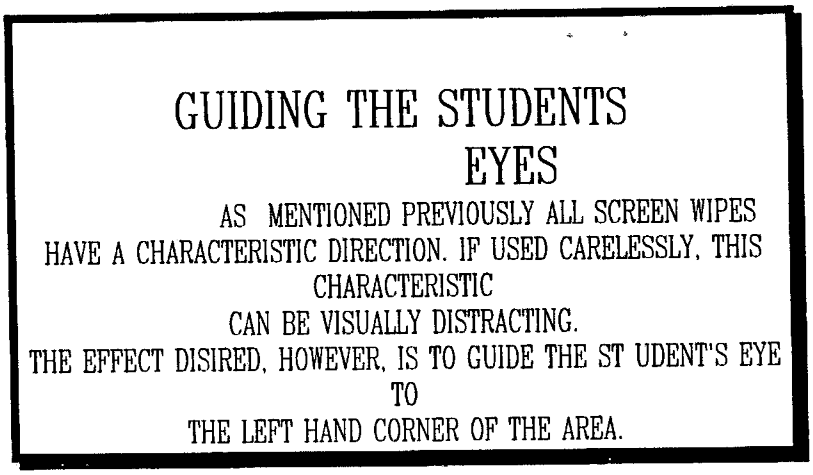
Research on human learning **(Fleming & Levie 1978)** indicates that dual modalities tend to enhance learning. They were referring to *dual sensory modalities* (visual,auditory and tactile), while **Rigney & Lutz (1976)** indicates that combined visual modalities also enhance learning.

The length of presentation is also a major factor in a tutorial. The more complex the information is, the more important it is necessary to break it into steps. Mature students can generally deal with longer presentations without losing interest or becoming confused.

A mitigating consideration is that longer presentations may not decrease motivation if they contain pictures, sound or animation. However, long sequences with a variety of text and graphics may still overload memory**(Alessi & Trollip 1985)**.

TEXT

Scrolling the texts on the display is not advisable, because most people find it difficult to readtext when it is scrolling, and, it is hard to demarcate between old information and new.

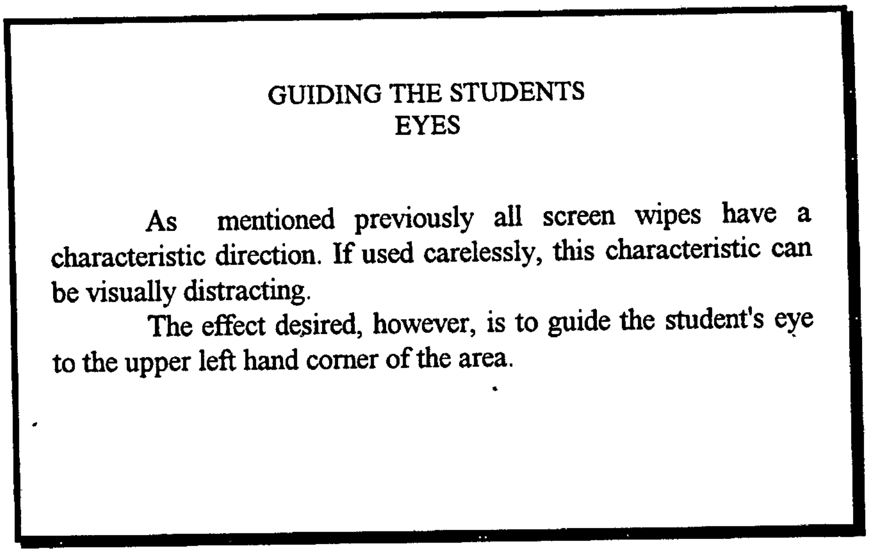


GUIDING THE STUDENTS
EYES
AS MENTIONED PREVIOUSLY ALL SCREEN WIPES
HAVE A CHARACTERISTIC DIRECTION. IF USED CARELESSLY, THIS
CHARACTERISTIC
CAN BE VISUALLY DISTRACTING.
THE EFFECT DISIRED, HOWEVER, IS TO GUIDE THE ST UDENT'S EYE
TO
THE LEFT HAND CORNER OF THE AREA.

Figure 2-5. A poorly formatted text.

Sentences and paragraphs should be well formatted on the display. Text should not be squeezed into half of the display leaving the half almost empty.

Text composed of all upper-case letters is more difficult than normal text which is primarily lower-case**(Alessi & Trollip 1985)**.



GUIDING THE STUDENTS
EYES

As mentioned previously all screen wipes have a characteristic direction. If used carelessly, this characteristic can be visually distracting.

The effect desired, however, is to guide the student's eye to the upper left hand corner of the area.

Figure 2-6. A well-formatted display.

Figures 2-5 contains text that is all in upper case, single spaced, crowded to the right side of the display, splits words across lines, uses inconsistent paragraphing conventions,and ends the page in the middle of the sentence. Figure 2-6 corrects all of these errors, leaving a display much easier to read.

GRAPHIC

Nowadays, microcomputer softwares and hardwares make graphic easy in any form of presentation. There are shareware programs which are much cheaper than softwares and as powerful as software.. Hardware such as *hand-scanner, vedeo frame-graber* are available everywhere with reasonable price. However, the effect of excessive use of graphic is sometimes negative. **Dwyer(1978)** presents considerable evidence that, when properly used, pictoral information enhances learning.

There are three primary uses of graphics during the presentation of a tutorial. They are :- (a) *primary information* in a presentation, (b) an *analogy or mnemonic* and (c) a *cue*(**Alessi & Trollip 1985)**.

COLOR AND ITS USE

As with graphics, softwares make the use of color increasingly easy. Color should not be misused, otherwise it will be detrimental to learning. Some colors, especially, yellow and green, are easier to perceive than others **(Durret & Trezona 1982)**. Red and blue is the most difficult colors to perceive.

One should avoid the combination of:-

- red and green,
- blue and yellow,
- green and blue, and
- red and blue.

The use of color in a tutorial should be consistent with common usages in our society. To use green to mean "**stop**" and red for "**go**", for example, will cause confusion and will result in errors by students.

FOCUSSING ATTENTION & TEXT QUALITY

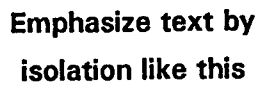
Enhancement:- The more difficult a tutorial is, the more readings involved, and the less frequently a student is required to interact, then the greater necessity to enhance the text on display. The following methods can be used to enhance the text.

Size. various sizes of text can be displayed by programming, authoring language or by softwares.

Inverse. Inverse writing is black letters on a white background, in contrast to the usual white on a black background. Inverse writing makes text stand out, much like using a yellow text-marker in a book.

Alternate. Typeface. Alternate typefaces or alternate fonts, include italics and bold characters.

Arrows and Boxes. Arrows can be used to point at important information, or information may be placed inside a box.



Emphasize text by isolation like this

Figure 2-7. Emphasizing text by isolation.

Isolation. An effective way of focussing attention on information is to display it with nothing else on the screen (Figure 2-7).

Blinking. The most dramatic device for focussing attention on text is blinking, in which words flash on and off or switch between normal and inverse writing.

Leanness. **Burke(1982)** uses the term leanness to describe an important quality of a tutorial. The program should say just enough to reach what is desired, and no more. This applies not only to the text descriptions but also to examples of concepts, sample applications of rules, pictures for demonstration purposes, and so on.

A practical consideration that reinforces the need for lean program is the limited capacity of computer displays.

Transitions. Maintaining a clear flow of ideas in a computer tutorial is more difficult than in textbooks. The limited display capacity requires changing pages more frequently. This can be minimized by using clear transition statements as mentioned below.

Press **<RETURN>** to continue.

Press **<RETURN>** for **Module 2: Use of equipments.**

Clarity. A lesson should avoid ambiguity, vague language, and should use consistent terminology. A word or phrase is ambiguous when it can mean more than one thing.

PROMPTS AND HELP

Prompts or cues are used to guide the student and to give hints. Prompts are displayed by underlining important words, or highlighting particular part of a graphic presentation.

Prompts are classified as either *formal* or *thematic*(**Markle 1969**).

Formal prompts aid the student by pointing out the form of the answer to a question the student should take. For example, formal prompt tells the student that "the answers to future questions are the underlined words".

In contrast, a thematic prompt is one that tells the student about the theme, or topic, of the answer. Following an incorrect answer, the hint "Try again, the answer has to do with distress" is a thematic prompt. Thematic prompts generally help the student understand information whereas formal prompts generally aid recall.

There are two types of help: *procedural* and *informational*. Procedural help is essential and refers to help in operating the lesson, such as changing pages. This information is usually provided in the directions for the lesson, so a common type of procedural help simply sends the student back to the directions.

Informational help assists with the content being studied. When help of either type is provided, it must be easy for students to access.One method is to allow the student to press the question mark<?>. Another is to allow the student to go at any time to a menu for help, directions, and the other choices by pressing <ESC>.

QUESTIONS AND RESPONSES

In tutorials, the most common method of interaction is to pose questions that the student must answer. They keep the student attentive to the lesson, force the student to practice, and assess how well the student remembers and understands the information.

Questions should occur frequently, and lengthy information presentations are best divided with interspersed questions. This requires the student to read or inspect small amount of information and then answer a question, which aids comprehension and recalls. The more the student interacts with the program in this way, the more attention will be maintained, the more the student will enjoy the lesson and the more learning will be facilitated.

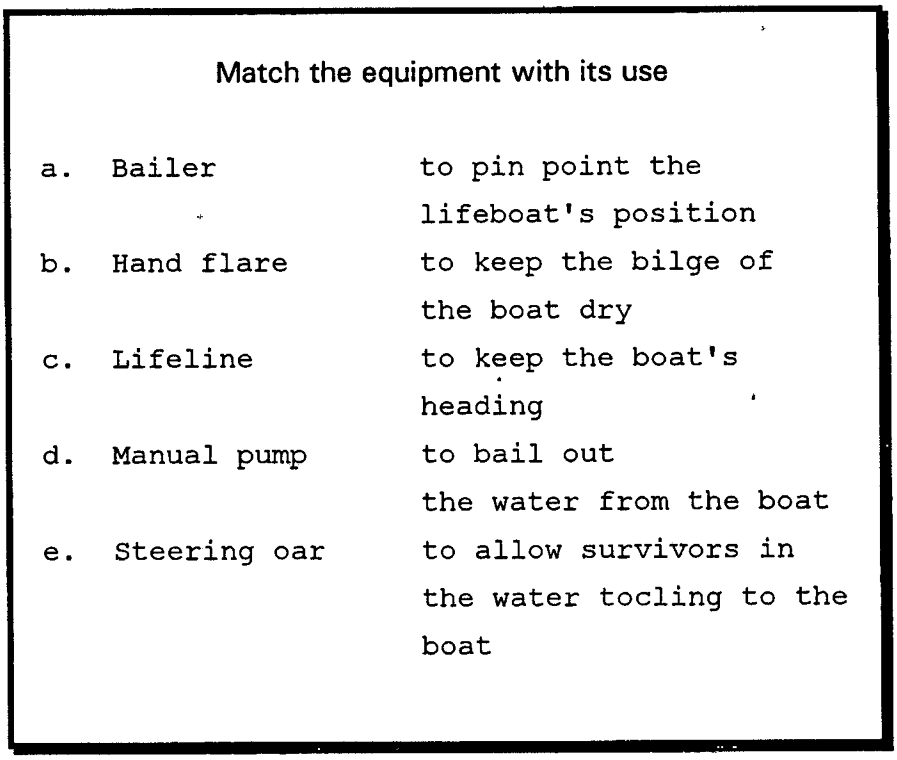
There are two types of questions in CBI; *alternate-response question* and constructed-response question.

ALTERNATE-RESPONSE QUESTIONS

These include *true or false* questions, *matching* questions, *multiple -choice* questions and *marking* questions. Most of the experts recommend the use of marking & multiple choice questions.

True-or-false questions are unreliable and the student has a 50-50 chance of answering correctly by guessing **(Rowntree 1990)**. A correct response, therefore, is not an accurate indication that the student has learned the material.

The matching questions tend to be complicated in format even if the information tested is simple. Here is an example screen of matching question(Figure 2-8).



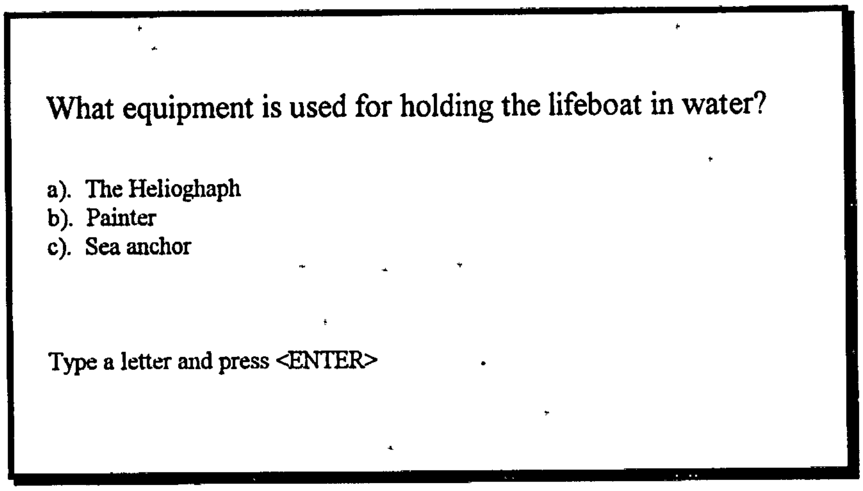
Match the equipment with its use

a.	Bailer	to pin point the lifeboat's position
b.	Hand flare	to keep the bilge of the boat dry
c.	Lifeline	to keep the boat's heading
d.	Manual pump	to bail out the water from the boat
e.	Steering oar	to allow survivors in the water tocling to the boat

Figure 2-8. A matching question.

MULTIPLE-CHOICE QUESTIONS

Multichoice-questions are the most common type of alternate question. Multiple-choice questions may use letter keys (Figure 2-9), number keys or a moving cursor to select the answer. Numbers are easier to find on a keyboard, but it is easier to press the wrong one. When the answers themselves are numeric, lettered answers are less confusing.



What equipment is used for holding the lifeboat in water?

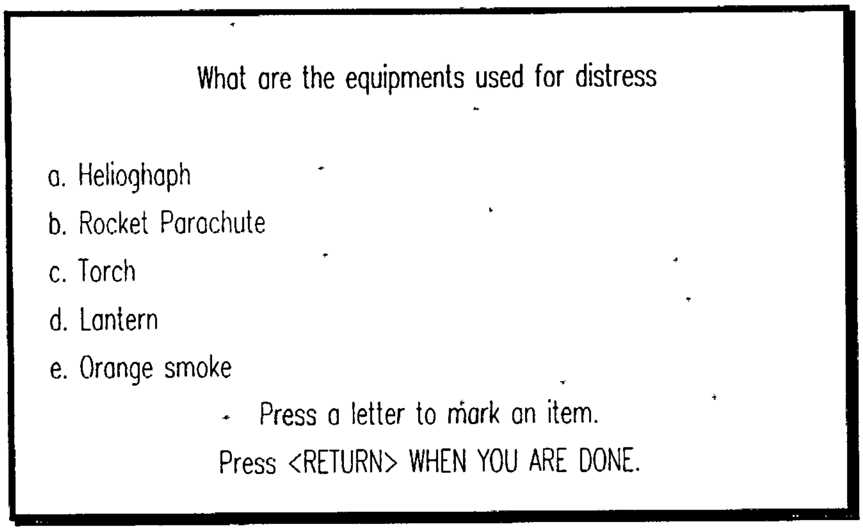
a). The Helioghaph
b). Painter
c). Sea anchor

Type a letter and press <ENTER>

Figure 2- 9. A multiple-choice question with letter choices.

MARKING QUESTIONS

Marking questions require the student to respond by marking parts of the display. The question in Figure 2-10) is like a multiple-choice question with more than one correct answer.



What are the equipments used for distress

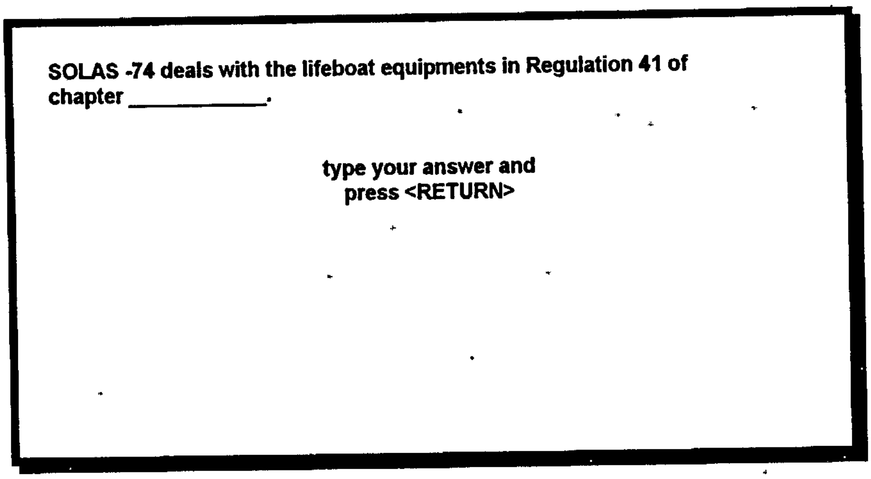
a. Helioghaph
b. Rocket Parachute
c. Torch
d. Lantern
e. Orange smoke

Press a letter to mark an item.
Press <RETURN> WHEN YOU ARE DONE.

Figure 2-10. A marking question.

CONSTRUCTED-RESPONSE QUESTIONS

The three major types of constructed-response question are: completion, short-answer, and essay questions. However, only the first two are used in CBI.



SOLAS -74 deals with the lifeboat equipments in Regulation 41 of chapter ____________.

type your answer and
press <RETURN>

Figure 2-11. A completion question.

Completion Questions:- have one or more missing words that the student must fill in. The primary considerations for completion questions are the importance of the missing words, and their location.

There should not be too many blanks in a question. With many blanks, the entire meaning of the statement is lost. The blanks should be near or at the end of the question.

Short-answer Questions:- require the student to type words or numbers. A single word response is more easier for a computer program to judge than the multiple-word response.

OTHER FACTORS AFFECTING QUALITY

(1) Using graphics in questions

There are two ways of using graphics in questions: as the context of the question, and as a hint or prompt.

(2) Question Relevance

Questions should be relevant only to the important information. If the student is to learn general concepts, questions about general concepts must be asked.

(3) Placement of Questions

Question may appear before or after the information assessed. Questions prior to presentations are rhetorical questions that motivate the student to search for the answer. From research work done by **Anderson & Biddle (1975)**, it may be beneficial to use questions in both ways; prior to presentations to focus attention information, and afterwards to enhance attention and provide practice.

JUDGING RESPONSES

Judging is the process of evaluating a response in order to give feedback, to make lesson sequence decisions, and to store performance data. A teacher is capable of:

(a) hearing or seeing a response and picking out the important information,

(b) ignoring extra words,

(c) recognizing synonyms,

(d) noting spelling errors, and finally,

(e) making a rational decision that the response is correct or not.

It is much more difficult for the computer to engage in this kind of intelligent judging, but it is the goal that lessons should strive for.

CONSIDERATIONS FOR JUDGING

(1) Length: - This is the length limit of an answer that the student should type for a question or questions. A limit should be set on the permissible length of a student response. For a sentence-completion question, the length limit should be 10 or 15 letters and should only be a single word. A

short-answer question might allow 30 or more letters. In all cases, if the student exceeds the permissible length, the program should not judge the response incorrect, but should inform him of the error and await a response of correct length.

(2) Time limits: - Normally, a certain time limit is set for a response that the student should make. **Alessi & Trollip(1985)** consider that lengthy response time may represent a problem, such as the student not knowing what to do. They recommend to have a program asking the student if help is needed rather than judging the lack of response to be incorrect.

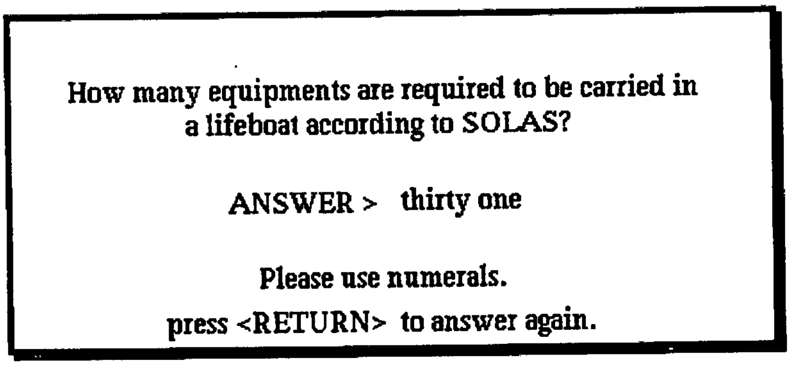
(3) Help and Escape Option:- Whenever the student is at a question, there should be provisions for the student to ask for help and to leave the program. Consequently, the judging machinery must recognize certain key presses or words not as responses, but as requests for help or requests to leave.

PROVIDING FEEDBACK ABOUT RESPONSES

The primary function of feedback is to inform the student about the appropriateness of a response that he has made. Following correct responses, it may provide reinforcement for the student. Following incorrect responses, it should provide correction, with the purpose of improving future performance.

TYPES OF FEEDBACK

Feedback following format errors. A format error is an error of form rather than content, such as using letters instead of numbers. Feedback should prompt the student to correct the format and try again. Please see the following figure 2-12.



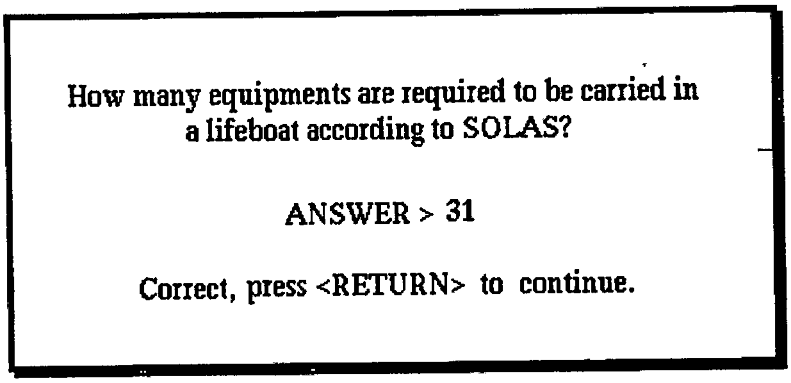
How many equipments are required to be carried in
a lifeboat according to SOLAS?

ANSWER > thirty one

Please use numerals.
press <RETURN> to answer again.

Figure 2- 12. Feedback for a format error.

Feedback following correct responses. When a response is correct, a short statement affirming this is made. This is usually done with a single word, such as good, or correct. Many programs randomly select different correct words for the sake of variety.



How many equipments are required to be carried in
a lifeboat according to SOLAS?

ANSWER > 31

Correct, press <RETURN> to continue.

Figure 2-13. Feedback for a correct response.

Feedback following neutral responses. A response may be neither right nor wrong, as when the student's name is asked. Feedback such as "Thank you, Press <RETURN> to continue " is appropriate in this case.

Feedback following content errors:- A response may be incorrect or only partially correct. Feedback following errors has a great effect in the success of instruction.

OTHER CONSIDERATIONS FOR FEEDBACK

POSITIVE & CORRECTIVE FEEDBACK

Feedback should be positive, should avoid negative statements and sarcasm, and should never demean the student. Jokes should not be made at the student's expense. While some students find this humorous, not all the student's will **(Alessi & Trollip 1985)**. Some students are more interested to see what will happen if they make errors. Therefore it is important to correct errors, but the corrections should not be so entertaining that they reinforce errors.

TIMING OF FEEDBACK

This refers to whether feedback should occur immediately after an error or should be delayed. The proper timing depends on the nature of what is being learned. It is likely that learning procedural knowledge **(J.R. Anderson 1982)** will be enhanced more by immediate feedback. **Gaynor(1981)** found that it can happen in many cases students believe the computer is not working properly when feedback is delayed.

DISPLAY OF FEEDBACK

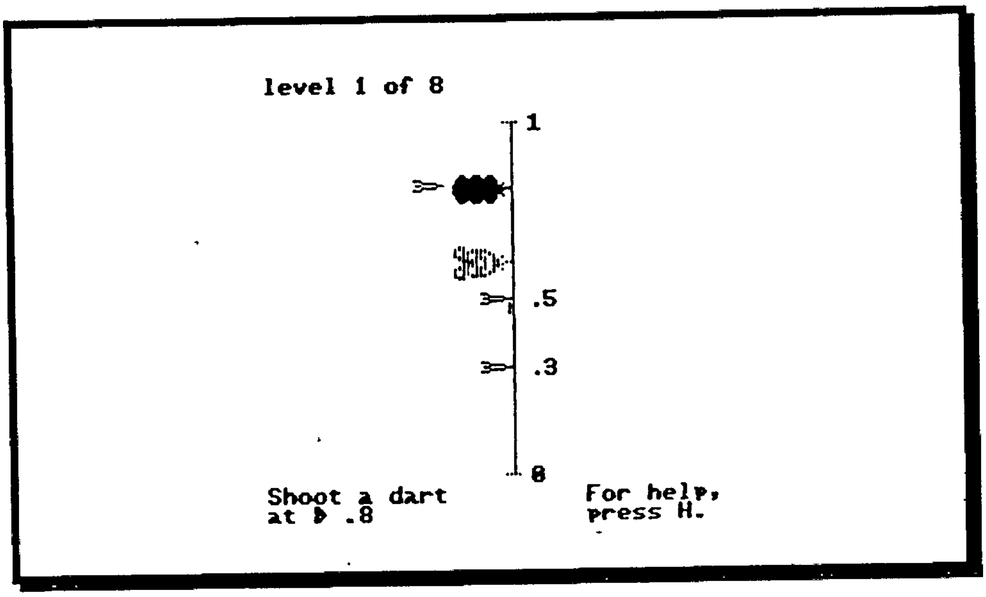
Text feedback. The most common type of feedback is to give the correct answer in text form, below the student's incorrect response. Text feedback need not supply correct answer. It may supply a hint so the student can try again. The common ways of providing a hint are:-

⇒ Rewording the question or problem, highlighting keywords or parts,

⇒ Showing the solution for a similar problem, and

⇒ Giving the student part of the answer.

Graphic feedback. Figure 2-14 shows a display from a program called *Decimal darts* in which the student must identify points on a number line by indicating the numbers where ballons are located. When the student gives a correct number a dart flies across the screen and pops a balloon. If the student gives an incorrect number, a dart flies across the screen and sticks in a part of the number line where there is no balloon.



level 1 of 8

Shoot a dart at ▶ .8

For help, press H.

Figure 2-14. Graphic feedback . Answering correctly pops a balloon(Courtesy of Control Data Corporation).

REMEDIATION

It refers to more extensive presentation of information for the student who is not learning the material. The types of remediation used on lessons are those used by most teachers in the classroom instruction.

Providing new information or repeating old information with more detail also aids the student who was paying attention but who did not understand the initial presentations. This may take the form of more examples, pictures, sample problems, or practice with simpler parts of the material.

Therefore, a remediation technique is always necessary for the computer lessons. It can be textbooks, workbooks, video or work with a teacher or colleagues.

SEQUENCING LESSON SEGMENTS

Some people call this flow-charting the programming and some say structuring a course.

It is a strategy to sequence a course or lesson and it involves the following considerations.

LINEAR OR BRANCHING

Linear sequence is the simplest and it progresses from one topic or concept to the next, first presenting information and asking questions.

When planning Linear Sequence, one should consider *hierarchy, familiarity* & *difficulty* of the information.

The following features of Linear Programming are found in **Skinner's theory**#1 of learning:

1. Small steps. Learners are exposed to small amount of information, to the next in an orderly fashion.

2. Overt responding. Students are required to respond overtly so that their correct responses can be corrected.

3. Immediate feedback. Immediately after making responses, students are informed as to whether or not they are correct. This feedback acts as a reinforcer if the answers were correct and as a corrective measure if the answers were wrong.

4. Self-pacing. Students proceed through the program at their own pace.

There are a number of variations of the Linear program. For example, some provide students with the opportunity to skip information that they are already familiar with. This procedure usually involves giving the students a pre-test on a certain section of the program, and if they perform adequately they are instructed to advance to the next section.

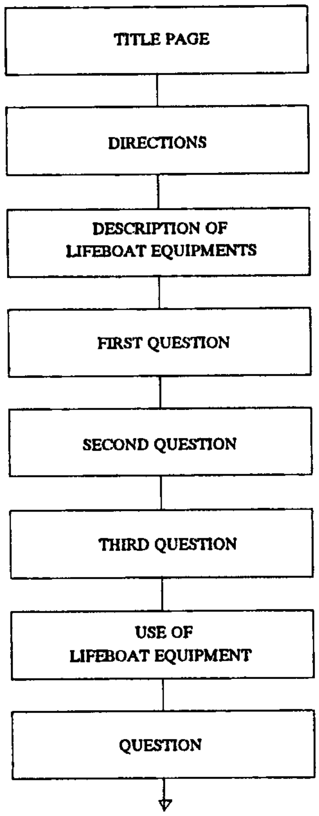


Figure 2- 15. Sequence in a linear tutorial.

Branching programming provides more flexibility by means of multiple choice format. It attempts to diagnose the

In figure 2-15, a common linear sequencing of a tutorial is described as a flow chart.

Branching sequence and amount of branching in a tutorial lesson vary with the student's individual performance.

Another consideration is the direction of branching: forward, backward or sideways. In figure 2-16, Branch 1 is a forward branch. If the student chooses to go directly to the second module, the information and questions in the first module are skipped.

Branch 2 is a backwards branch. If performance on the three questions was not adequate, the student is returned to repeat the information and answer the questions again.

Branch 3 is a sideways branch. If the student answers questions incorrectly, new information is presented before the question is repeated.

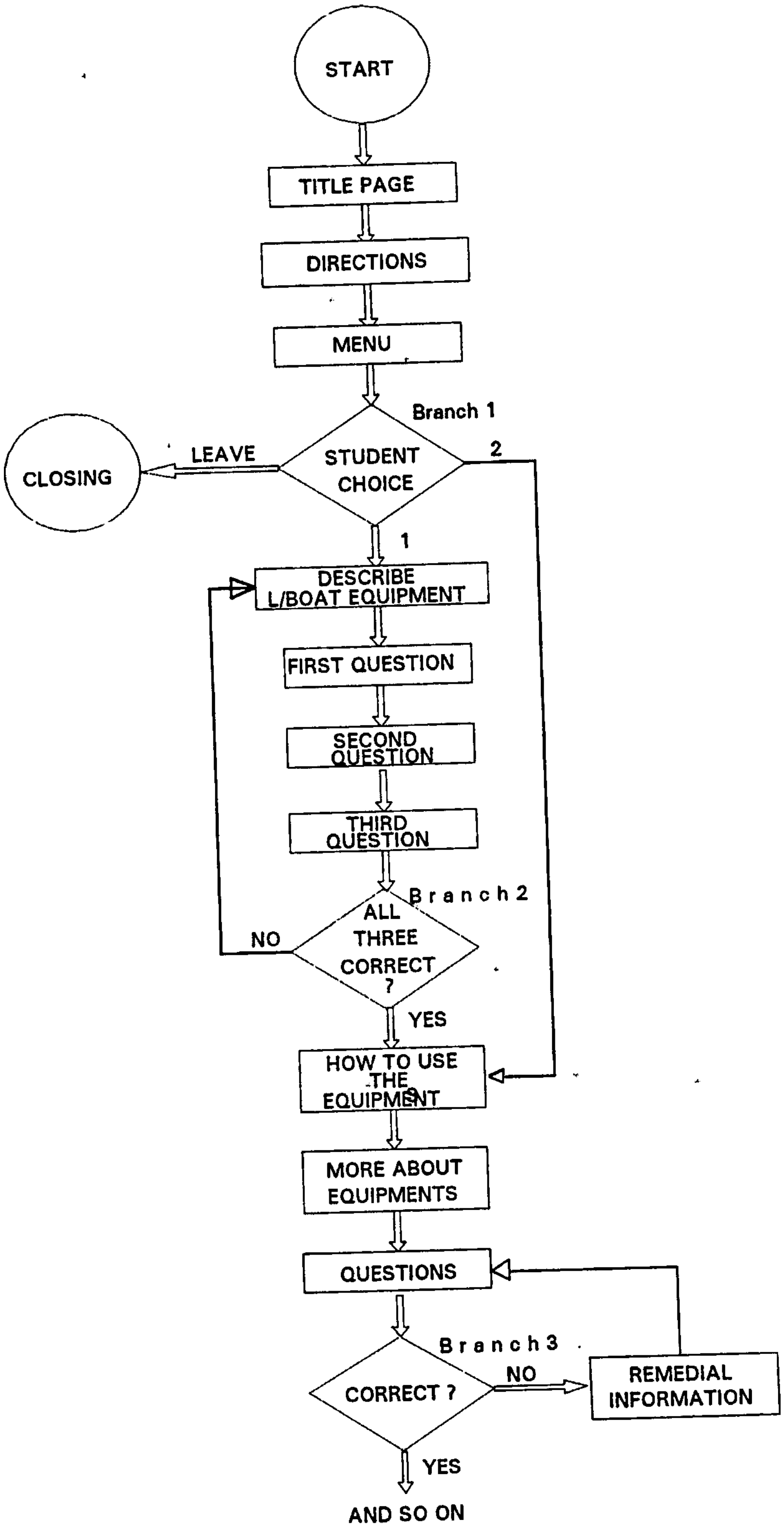
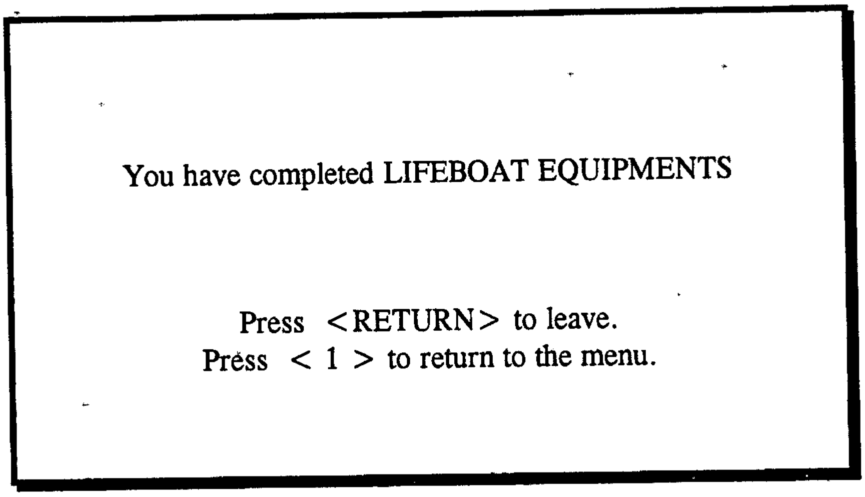


Figure 2-16. Sequence in a branching tutorial.

CLOSING A TUTORIAL

There should be two ways to end a tutorial; temporarily and permanently. An option should be provided for the student to leave the lesson temporarily. The permanent ending will occur when the lesson is completed successfully.

For both types of ending, data may need to be stored and one should make clear that the student is leaving the program. In temporary ending, the student should be given a message indicating how to return to the point where he left off.



You have completed LIFEBOAT EQUIPMENTS

Press <RETURN> to leave.
Prèss < 1 > to return to the menu.

Figure 2-17. The final message.

It is practical to provide summary statements about the student's performance, if recorded, the important points of the lesson and the recommendations for further study.

DRILLS AND PRACTICE

The purpose of CBIs in a drill is not to teach the student. They are only for practicing the knowledge that has been gained by other applications such as tutorial, simulation, reading from textbooks, a classroom lesson, or a group discussion.

Drills are applicable to any type of learning, provided that initial presentation and guidance have already occurred. Examples for drills are spelling or foreign language translation, verbal information, arithmetic facts, problem-solving in the physical and social sciences(**Alessi & Trollip 1985**).

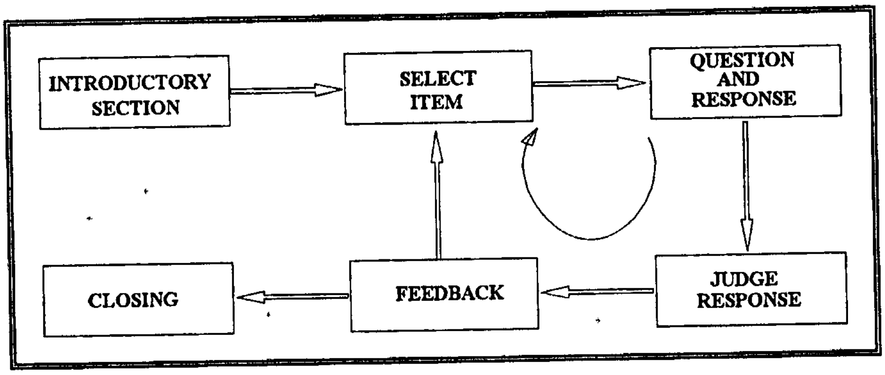


Figure 2-18. The general structure and flow of a drill.

Figure 2-18 shows the general procedures of a drill. There is an introductory section followed by a cycle that is repeated many times. Each time the cycle is repeated the following actions generally take place.

(1) An item is selected.

(2) The item is selected.

(3) The student responds.

(4) The program judges the response.

(5) The student receives feedback about the response.

After a number of items selected, the program eventually terminates. There are variations in selecting items and terminating the program.

Use of text, graphics and sound are the same as in tutorial. As in tutorials, drills have questions but they are not called questions. They are called items.

TYPES OF DRILL

The drills can be divided into two types; namely, *Intelligent* and non-*Intelligent* drills.

An Intelligent drill is one that responds to the learner as if a teacher or tutor who possesses expertise in these various areas would respond. The progressive state drill, the variable item interval drill, and concept learning drill are intelligent drills. Drills that simply follow the same preset sequence for each learner are considered as non-intelligent drills.

STRATEGIES OF A DRILL

DIFFICULTY, FLUENCY AND PACING

In drills, the program must create difficulty to the learner. How difficult is to be considered although it is not very important. There are many ways to create difficulty and some of them are as follows.

(1) keep difficulty constant with the items.

(2) Increase difficulty continuously and gradually based on student performance.

(3) Group the items by difficulty.

Fluency and speed play a great role in drills. Speed can be facilitated by using items that do not take long to read and answer.

Pacing means giving the student a limited amount of time to respond. Depending on the type of drill(whether its goal is speed or accuracy), the student should have enough time to read the information.

PREPARATION

Preparation is time-consuming and consists of the following tasks.

(1) Item preparation.

(2) Item selection.

ITEM PREPARATION

Two methods is normally used to present items. They are as followed.

List selection. Most drills, such as vocabulary, translation, spelling, multiple-choice questions, single-word response questions, and science problem solving select items from a list. All the items that may be presented to the student are constructed ahead of time by the author and put into a list, or library, in the computer's permanent storage. The drill program then selects items from the list and presents them to the student. Answer keys are also stored in the same list.

By an algorithm. Drills involving mathematics frequently use an algorithm to generate items. An algorithm is a procedure(a set of rules) to produce an item. Figure 2-19 shows a simple algorithm for producing an *addition facts drill.*

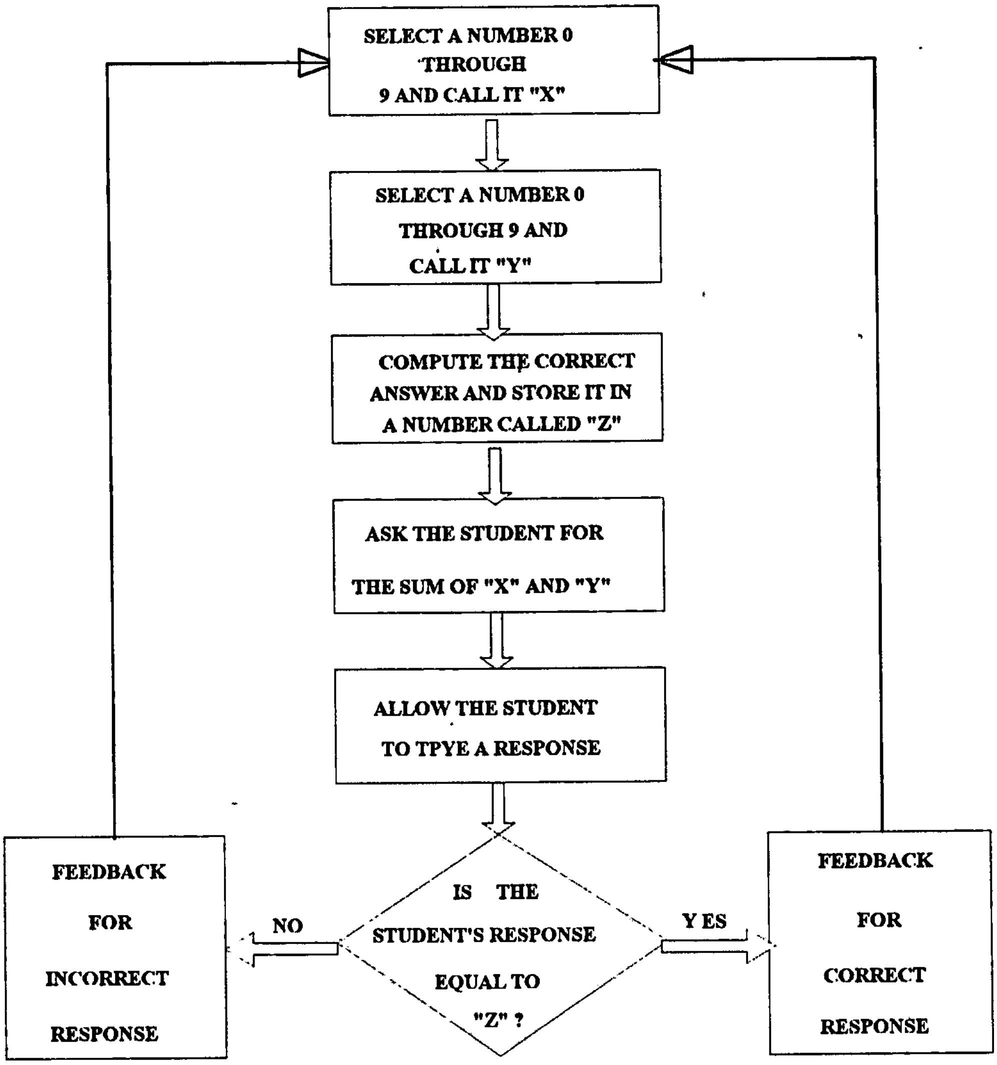

SELECT A NUMBER 0 THROUGH 9 AND CALL IT "X"
SELECT A NUMBER 0 THROUGH 9 AND CALL IT "Y"
COMPUTE THE CORRECT ANSWER AND STORE IT IN A NUMBER CALLED "Z"
ASK THE STUDENT FOR THE SUM OF "X" AND "Y"
ALLOW THE STUDENT TO TPYE A RESPONSE
IS THE STUDENT'S RESPONSE EQUAL TO "Z" ?
NO
Y ES
FEEDBACK FOR INCORRECT RESPONSE
FEEDBACK FOR CORRECT RESPONSE

Figure 2-19. A simple item generation algorithm(Alessi & Trollip, 1985).

According to the figure,

(a) It first selects two numbers and calculates their sum.

(b) It presents the numbers to the student and asks the student to type the sum.

(c) The student responds and the program compares the number typed to the number calculated internally.

(d) The program indicates if the student was correct or not.

(e) The procedure then repeats and administers another addition problem with different numbers.

Program can be a combination of lists with algorithms. For instance, one can make a drill for *the correction of altitude for sun* in which it contains lists of items like (a) sextant altitudes , (b) index errors, (c) GMT date and time, (d) dead reckoning position, and algorithm for generation of true altitudes. During the drill exercise, an item type is selected and the corresponding algorithm generates questions.

ITEM SELECTION PROCEDURES

After setting up a list or algorithm for the items, one must consider how to select the items. The principle of selection items should be based only on the student performance. There are two main methods of selecting items and they are briefly explained as followed.

Random Selection. Each iteration of item or generation of algorithm is selected randomly from a list. During the program, when the student answers incorrectly, he will be told the correct answer, but he may forget it before it is selected and presented again. The fact is that the student should get more practice on items causing difficulty than on those whose answers are known.

Organized Queuing. This method brings solution for the above mentioned short-coming. Queuing means determining ahead of time the order that items will be presented. There are two methods of queuing, namely; *flashcard queuing* and *interval performance queuing*.

Flashcard queuing does the same way as people play flashcards. Suppose, there is a deck of flashcards with French words on one side and English equivalents on the other, the procedure of using these cards will be as followed.

(1) Look at the top card and translate it.
(2) Look at the other side and, if correct, discard it from the deck.
(3) If a word is translated incorrectly, that card will be placed at the bottom of the deck.
(4) When all cards are discarded, shuffle them and begin again.

Computer can be programmed to select the items the same way as we do in flashcards. However, in this method, especially at the start of the program, there is a long gap between the time the item is discarded and the time it appears again. Thus the student is likely to forget the response before it is presented again.

Variable Interval Performance Queuing. The basic principle is that when an item is answered incorrectly it is positioned or queued at a number of new positions in the future order of presentations, or *future queue*.

The following flowchart in figure 2-20 shows a simple general principle of the Variable Interval Performance Queuing (**VIP**).

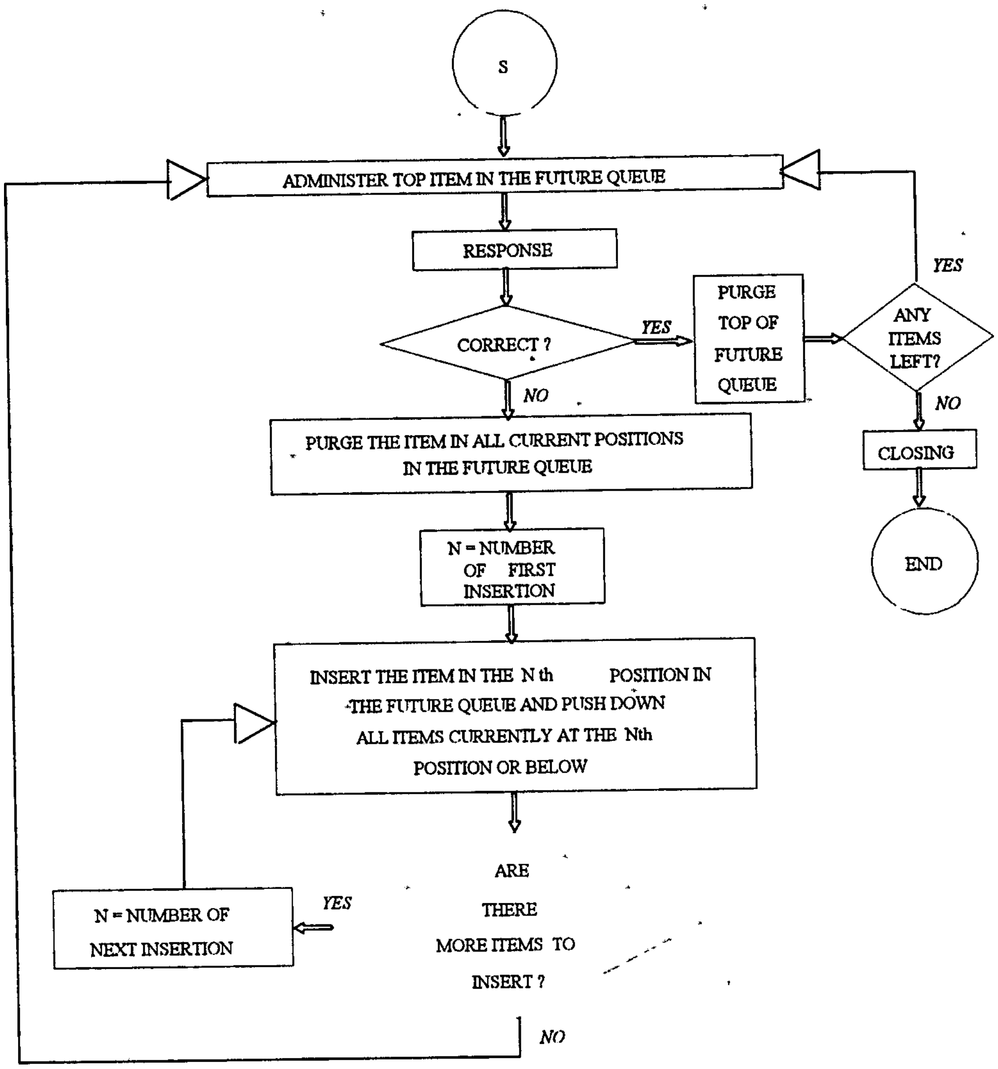

S
ADMINISTER TOP ITEM IN THE FUTURE QUEUE
RESPONSE
CORRECT ?
YES
PURGE TOP OF FUTURE QUEUE
ANY ITEMS LEFT?
YES
NO
CLOSING
END
NO
PURGE THE ITEM IN ALL CURRENT POSITIONS IN THE FUTURE QUEUE
N = NUMBER OF FIRST INSERTION
INSERT THE ITEM IN THE N th POSITION IN THE FUTURE QUEUE AND PUSH DOWN ALL ITEMS CURRENTLY AT THE Nth POSITION OR BELOW
ARE THERE MORE ITEMS TO INSERT ?
YES
N = NUMBER OF NEXT INSERTION
NO

Figure 2-20. VIP queuing.

By this procedure the items the student finds difficult can be repeated frequently.

TERMINATION

The decision to end a drill is called termination. Like tutorial, a drill can be terminated temporarily and permanently.

Temporary termination can be provided by assigning a key, usually **<ESC>** key. Direction for termination should be given at the beginning of the drill.

Permanent termination should be based on student performance. It should be indicated by a message saying that the student has mastered all items and now has the option of terminating it permanently.

FEEDBACK

Feedback in tutorials applies to drills as well. The additional factor is to give a clear message after the student has made errors caused by confusion between similar items, in other words, *discrimination error*. For instance, mistaking a rectangle for a square may be a discrimination error.

In a foreign language translation drill, it is hard to program the computer to recognize how similar two words are in spelling. It is very difficult to program the computer to recognize that two words sound similar.

Following a regular error, a drill will indicate the response was incorrect and give the correct response. However, following a discrimination error, one should not only indicate that the response was incorrect and what the correct response is, but should also point out that the response given is the correct response to another stimulus in the drill.

MOTIVATING THE STUDENT

The motivational quality of drills is low, because they are repetitious in nature, asking the same or similar questions, requiring the same response from the student and giving the same type of feedback. One should consider to enhance motivation. **Alessi & Trollip (1985)** pointed out the following methods to increase student's motivation.

COMPETITION AGAINST OTHER STUDENTS

Two students may use a drill simultaneously, or the computer may store data on a number of students so that each can see their performance compared to others. This is done by showing a rank-ordered list of student performance. This method is found in a drill program for practicing typing by computer keyboard, called typing tutor.

The problem with competition among students is that while it serves as a powerful motivator for students who do well, it may be a punishment and embarrassment for those who do poorly.

COMPETITION AGAINST THE COMPUTER

This method has the advantage of allowing all students to succeed and receive reinforcement. It is also possible to adjust the difficulty of the computer's competitiveness so that against the best students, the computer is very competitive and against the poorest students it is least competitive.

In this way, the drill program can motivate each student to work at his or her best, and when the student does so, he or she will achieve success by beating the computer. The problem here is that it is not suitable in some cases such as a spelling drill.

DATA STORAGE

In drills, data may be stored on temporary basis (RAM) or permanent basis like computer disks or other magnetic media. Temporary storage is *volatile* and permanent is *non-volatile*.

The decision on which data should be stored permanently and which data be stored temporarily depends on many factors. The followings are a few factors to be considered.

(a) If a simple random item-selection procedure is used, no data is necessary. If VIP queuing technique is used, the author must store data defining the future queue. However, the data is needed to be stored temporarily, because it is constantly changing.

(b) The other data necessary to be stored temporarily are the number of items presented, number of minutes in the drill, total correct or percent correct, number of items retired, and number of items left.

(c) Data that needs permanent storage are data about student performance, and data for restarting if the program is terminated temporarily. The data for restarting can be generated from the *future queue* at the time of termination. When the student restarts, the future queue is retrieved from the disk and placed back in memory(RAM) as temporary data for further use.

SIMULATIONS

Although simulation programs are the hardest programs to describe, they are widely used in maritime education.

Actually, they do not teach the students, in fact they show something. For example, a simulation of planets revolving around the sun, may be a useful supplement to an earth-science text book or a classroom demonstration.

TYPES OF SIMULATION

Simulation programs are varying from *laboratory-simulations to flight simulators, radar simulators, ship handling simulators* and so on. When studying simulation,

Alessi & Trollip(1985) mentioned that they can be divided into four main categories, namely;

- *Physical*
- *Procedural*
- *Situational*
- *Process.*

PHYSICAL SIMULATIONS

In this simulation, a physical object is displayed on the screen, giving the student an opportunity to use it or learn about it. Typical examples are a machine that the student must learn to operate, or some scientific laboratory equipment.

In a simulation program called **OOW Simulator**[#2] as shown in the figure 2-21, for example, the student has a full set of ship-handling controls and instruments just below the screen. The purpose of the simulation is to learn how to use these instruments to control the navigation of the ship and how to learn the rule of the road. It has a 3-D view, with a choice of view ahead, astern , port or starboard. In any of the four 3D views, the student can use binoculars and the bearing compass.

The student can touch the controls by pointing a mouse on the control and click it to activate. It has the following functions:

- ♦ Telegraph with eight steps from Full Ahead to Full Astern.
- ♦ Choice of manual or automatic steering.
- ♦ Choice of altering course to a given heading or altering a chosen number of degrees to port or starboard.
- ♦ Speed log, heading repeater, rudder angle indicator, clock and stopwatch.
- ♦ A whistle capable of producing the prescribed long and short blasts.

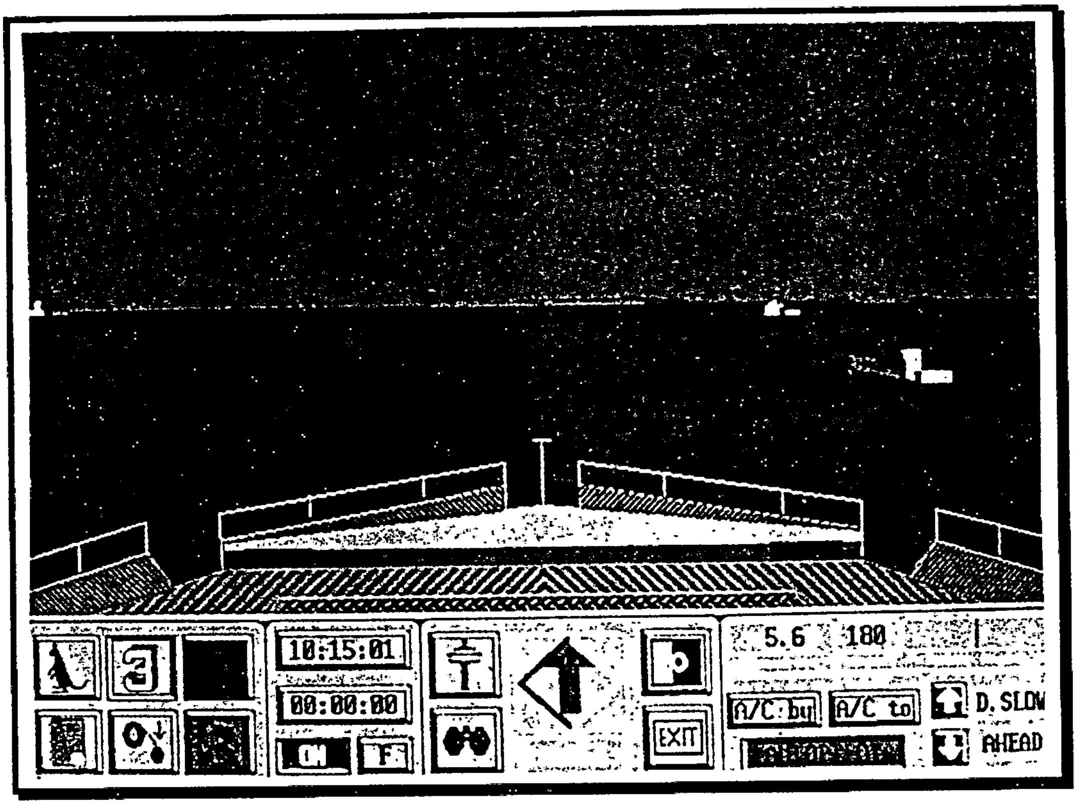


Figure 2-21. A Screen from the PC - based OOW Simulator.

Other maritime related simulators such as *Radar Simulator, Ship Handling Simulator, Replica Engine room Simulator,* etc. can be classed as Physical Simulation. The above mentioned **OOW Simulator** is a *PC-based simulator.* However, there are PC-based Radar simulation programs available now.

PROCEDURAL SIMULATIONS

Despite the fact that physical simulations are very common, in most lessons they play a secondary role to prrocedural simulations. In other words, the physical simulation exists only as a vehicle for the procedural content. Thus the student learns about how the simulated machine works, not as an end in itself, but rather as a mean for acquiring the skills and actions needed to operate it. In a radar simulator, the method and procedure of navigation with radar is the main objective of the course. To learn the procedure of how to use the control of a radar is a secondary objective.

SITUATIONAL SIMULATIONS

Situational simulations deal with the attitudes and behaviors of people in different situations, rather than with skilled performance. Unlike procedural simulations, which teach sets of rules, situational simulations usually allow the student to explore the effects of different approaches to a situation, or to play different roles in it.

The student is an integral part of the simulation taking one of the major roles. The other roles may be performed by students who interact with the same program, or by the computer playing the role of a person. Some types of games may also be classified as situational simulations.

PROCESS SIMULATIONS

Process simulations are different from other simulations in several important ways, The student neither participates in the simulation as with situational simulations, nor constantly manipulates it as in physical or procedural simulations. Rather, the student selects values of various parameters at the beginning of the simulation, and then the process occur without intervention.

Economists, for example, use process simulation to determine how much revenue would be available for appropriation, the economist would select values for the unemployment rate, gross national product, labor productivity, income tax rates, and so on. The simulation is then be run, and the result would be an estimate of how much revenue would be available, given the circumstances defined at the start. Learning from such simulations occurs by repeating the process a number of times with different starting values and comparing the results.

The Expert system such as **SEABEL** of Netherlands, is another example. It has a huge amount of data such as chemicals, types of ship, addresses to make contact in an emergency that occurs in the North Sea. Once an accident occurs, the program compiles a question list, and one may put the relevant data in the questions. Then after the program consults with the database, it will simulate the behavior of the accident such as:-

- * the behavior of an oil spill at sea , or
- * the behavior of a gas cloud above the sea,
- * determination of a search area for
  - - drown persons
  - - vessels
  - - floating objects.

It can generate the following relevant informations at the same time.

- # List of applicable response actions
- # List of threatened areas
- # Consulting databases such as,
  - • - addresses
  - • - platforms
  - • - coastal municipalities
  - • - pipelines
  - • - expertise and equipment in the Netherlands
  - • - response actions.

GENERAL STRUCTURE OF A SIMULATION

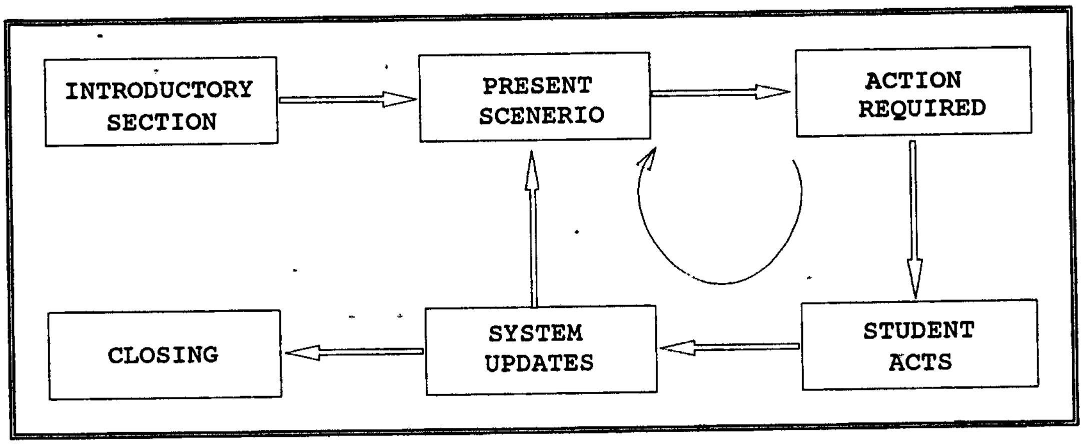


Figure. 2- 22 The general structure and flow of simulations.

As tutorials and drills, it has an introduction followed by a cycle, which is repeated. The occurrences in simulations are as follows.

(1) A scenario is presented.

(2) The student is required to react.

(3) The student reacts.

(4) The system changes in response to this action.

INITIAL PRESENTATION

Clear and complete directions are more important in simulations than in the previous applications, because students engage in activities that are more complicated and varied**(Alessi & Trollip 1985)**.

In most radar simulations, the title page and directions are provided only on the instructor's display unit. Therefore, the instructor is necessary to do *briefing* to the student before starting the scenario.

STRATEGIES OF A SIMULATION

CONTEXT OF THE SIMULATION

The context of the simulation is the system or phenomenon being simulated. It is the physical entity, the procedure, the situation of which the student is a part, or the process the program mimics. Each context is made up of a number of factors, which determines both its nature and the nature of the interactions with it. These factors are:

- The objects
- The precision
- The level of realism
- The relationship of the context to the instructional goals
- The sequence
- Whether the context is pictorial or textual in nature
- The location of the user
- Whether the user is primarily the actor or reactor

OBJECTS

The objects in the context are any physical entities, pictured or described. Examples of such objects are an airplane, chemical apparatus, a telephone, space ships, hospital patients, automobile engines, unknown substances, a job application, road signs, school principals, animals, corporations, and countries. Some simulations may deal with a single object, such as a piano, while others may deal with many, such as the various people in an elementary school.

Having a larger number of objects does not necessarily make the simulation more complicated, either to program or to use. Rather, it is usually the presence or absence of people as objects that increases complexity. Rules governing human behavior are far less understood than those governing the behavior of airplanes, pianos, and animals.

PRECISION

The presence or absence of people as objects in the context is related to the precision of the context. Precision refers to how well what is being simulated is understood. The most precise contexts are those involving strict mathematical, physical, or chemical laws. It is well known what happens when a distillation apparatus is heated, or when a five-pound weight is dropped from ten feet. However, even things that follow physical or chemical laws may have elements of probability or chance. That is, some of the factors that influence the object are either unknown or impossible to determine.

An automobile engine, for example, follows physical and chemical laws completely, but deciding why an engine runs poorly is still a difficult matter because there are so many physical and chemical influences on the many parts of the engine. There are also many unknown influences as well, such as the care the engine has received in the past, or how fast the owner normally drives the car. Thus the operation of an engine is based both on chance or probabilistic considerations and on scientific or mechanical ones. The more chance is involved, the less precise the context is and the harder it will be to program the simulation.

The extreme case occurs when people are involved. Very little is really understood about individual human behavior, which makes predicting it almost impossible. Simulations that include humans as objects, therefore, usually incorporate a great deal of chance. Consequently, they are the least precise and most difficult to program.

It is useful to keep in mind that simulations require a description or prediction of the behavior of the various objects in them. So, when trying to determine how difficult a simulation will be to program, think about how predictable the various objects are. Pianos, for example, are very predictable, because pressing a particular key will always result in the same sound. Automobile engines are less predictable. When you turn the ignition key on a cold morning, the engine may not start. People are quite unpredictable, although the degree of predictability varies from one individual to another. Thus, simulating a piano is easier than simulating an automobile engine, which in turn is easier than simulating a person**(Alessi & Trollip, 1985)**.

SEQUENCE

The sequence of the context refers to whether the events that occur do so in a linear, cyclic, or complete fashion. The events of a titration, for instance, are essentially linear. There is basically one way in which a titration should be performed. In Radar simulation and Ship handling simulation programs, the sequence is a mixture of cyclic and complex, as they involve a lot of actions and reactions.

Although complexity is a function of the context it is usually simplified in order to make the simulation easier to design and program, and to facilitate learning for the student.

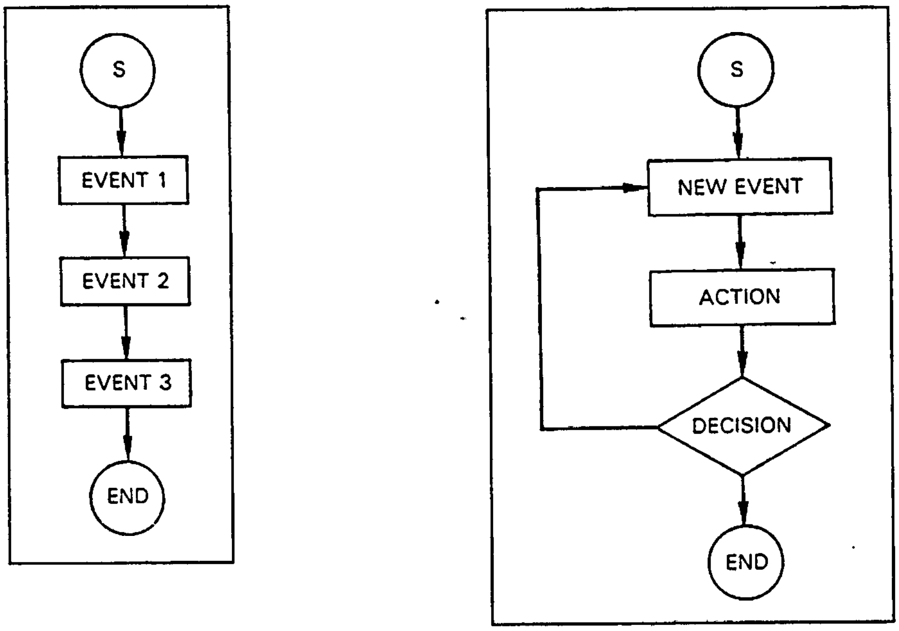


Figure 2-23. Linear and cyclic sequence in simulation.

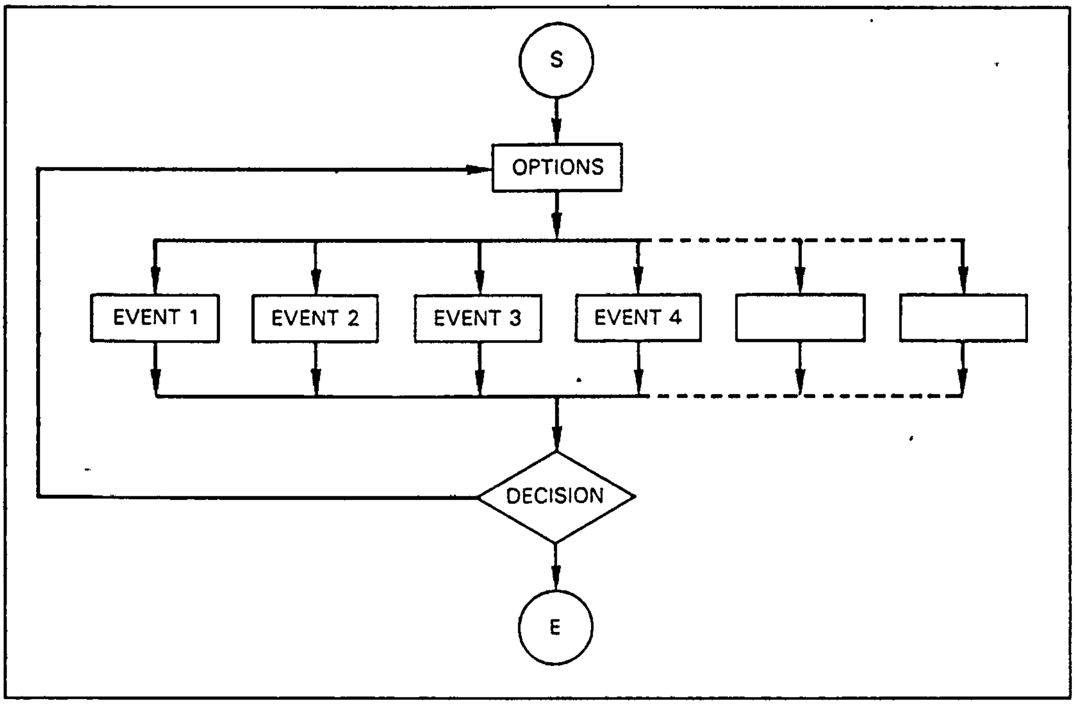


Figure 2-24. Complex sequence in simulation.

Other contexts, particularly procedural ones, have a single correct sequence of events. For example, in a radar simulation program, where the student has to bring his ship from one port to another across a Traffic Separation Scheme , in clear visibility, he must follow the procedures appropriate to Collision Regulations.

WHETHER THE CONTEXT IS PICTORIAL OR TEXTUAL IN NATURE

Physical contexts are usually pictorial in nature, situational contexts are usually textual in nature, and procedural or process simulations are commonly either or both. By pictorial or textual the manner in which the context is most easily depicted. It is easier to depict road signs by showing pictures of them than it is by describing them in words. Conversely, it is easier to depict the first year of teaching by describing it in a story rather than by trying to draw it. Depicting a context pictorially is usually more realistic but not necessarily better from a pedagogical point of view. Sometimes it is just not possible to use anything other than text.

LOCATION OF THE USER

The location of the user refers to whether the student using the simulation is considered to be one of the objects in the context, or is external to the context. Being a part of the context, however, does not necessarily mean the student is a person in the contest; the part may be an animal or physical object. In a situational simulation, the student is usually a part of the context. For physical, procedural, or process contexts, the student generally manipulates and observes the context from outside. In Radar and Ship handling simulations, it is advisable to have two students in one ownship; one acts as a navigator and the other as a helmsman. If there are more than one ownship, two groups can interact each other and they may become actor and reactor in the simulation.

WHETHER THE USER IS PRIMARILY AN ACTOR OR REACTOR

In some contexts the user is the primary actor to which objects in the context react. In contrast, the context may be the primary actor to which the student or user reacts. Sometimes neither one takes the primary role, but they act and react in equal ways. In Radar Simulations, the student is an actor who controls the movement of his ownship by engine telegraph and steering wheel. However, the instructor can manipulate the situation any time from his console unit.

PRESENTATIONS

TYPES

The four major types of presentations are usually all present in varying degrees in every simulation. They are:

- Choices to be made
- Objects to be manipulated
- Events to react to
- Systems to investigate

Choices to be made are usually textual because they involve the selection of one option from among many. Objects to be manipulated, such as the controls of an airplane or pieces of chemical apparatus, are frequently pictorial. Events to which the student must react can be of any mode. Thus, the student may be told that a patient's vital signs have deteriorated; a student pilot may see a change in the instruments; or a young musician may hear a note played by the computer. Systems to be investigated also typically use mixed modes.

FIDELITY

In the previous section we indicated that most simulations make compromises with respect to reality, usually through the process of simplification. The result of such changes is a decrease in the fidelity of the simulation. It is important to remember that such a decrease in fidelity does not necessarily mean that the effectiveness of the simulation has also decreased.

On the contrary, it is usually beneficial to simplify the context.

Fidelity has three main components: *completeness, realism,* and *timing.*

Completeness refers to the extent to which the important components of the real context exist in the simulation. All simulations simplify the context by eliminating objects or other components that are not essential to the instructional goals.

A flying simulation for a beginner will not present all the instruments nor will require interaction with air traffic controllers or other airplanes. Only those objects contributing to the task at hand would appear in the display. As proficiency increases and the simulation becomes more complex, other instruments may be added. At any stage the completeness of the simulation should be assessed against the instructional needs of the learner and the effectiveness of presentation, rather than merely against reality.

The realism of the simulation refers to the degree to which a particular component is like its real counterpart. Once again increased realism is not necessarily tied to increased effectiveness. In a program about weather, it may be important to draw a cumulonimbus cloud quite accurately, showing its anvil shape and giving details about growth rate, height, and intensity. Similarly, if the purpose of a simulation is to introduce chemical apparatus, it would be important to include details such as volume markings, stoppers, and correct size. If the simulation merely uses the apparatus as part of some experiment, simple silhouettes would suffice.

Simulations often compress or expand the time frame in which a phenomenon occurs. If the time frame of the real context is reasonable, then the simulation may follow it faithfully. However, most contexts require distortion of the time frame in order to be simulated.

STUDENT ACTIONS

MODE

Simulations incorporate more types of input modes than other instructional methodologies. In addition to the keyset, *game paddles* or *joysticks* are frequently used. Devices such as *touch panels, light pens, the mouse,* and *speech input and output* are also becoming more popular and may be seen in increasing numbers of simulations. The use of game paddles and touch panels is particularly useful in simulations that require the student to manipulate objects on the screen.

The keyset is easier for making choices at menus or for requesting information.

As always, the age and the typing ability of the student are major considerations.

Finally, the use of several modes probably enhances interest and stimulates more learning than would the use of a single mode **(Rigney 8c Lutz, 1976)**.

FIDELITY

Student actions, like presentations, have varying degrees of fidelity. This means that the similarity between the student's action and the action of a person in the real situation varies. In this context, fidelity may refer either to the mode of action or the type of action. Typing in the desired rotation speed of a motor is of lower fidelity than dialing a simulated knob to achieve the same end. Multiple-choice questions are not common in the real world as a means for diagnosing why an automobile will not start in the morning, whereas touching the buttons of a simulated telephone is similar to the actual activity.

SYSTEM REACTION OR FEEDBACK

NATURAL OR ARTIFICIAL

After taking an action, the student usually receives some type of feedback about the result of that action. In tutorials and drills, feedback is usually immediate, corrective, and positive. In simulations, this is frequently not the case because of the presence of a new factor, which is whether the feedback given is similar to the real world or similar to tutorials and drills. The former is called natural feedback, and the latter artificial. Which one is used affects the fidelity of the feedback**(Alessi & Trollip, 1985)**.

IMMEDIATE OR DELAYED

Regardless of the situation in the real world, simulations provide the option of giving feedback that is natural or artificial, or none at all. The main reason for artificial feedback is to give immediate, more understandable feedback. The advantages, on the other hand, of using natural feedback are that :

# it has greater face validity,
# is usually more interesting, and
# often results in better transfer of learning.

Immediate verbal feedback about actions helps correct them before the student becomes hopelessly lost and confused. Natural feedback is more like the real world, and it better prepares the student for performing in the real world.

In light of these considerations, our recommendation is to use immediate corrective feedback. In contrast, when a more advanced student uses the simulation, especially when it is used for practice or as a test, it should give natural feedback as much as possible, whether delayed or immediate.

A good simulation might start the student out with very helpful, immediate, and corrective feedback. As the student progresses and performance improves, the simulation could reduce the amount of artificial feedback, replacing it with more natural feedback. In general, student actions fall into four major categories:

# good or desirable,
# unnecessary or neutral,
# bad, and
# critical.

The feedback that is provided should be appropriate both to the nature of the student action and to the intention of the instruction. When a student action is necessary or good, which helps progress toward a goal, immediate feedback is least necessary, and natural feedback is suitable in most cases.

The examples above are meant only to illustrate the kinds of considerations involved in choosing feedback. They are not rules to be followed blindly.

Artificial, natural, delayed, and immediate feedback can always be used, and may be used in various combinations. Each situation must be analyzed to determine which type of feedback will best help students attain the objectives of the simulation.

SEQUENCE OF THE SIMULATION

The sequence of events in a simulation is a function of the sequence in the real context, although usually a simplification of it. A context with a linear sequence is the easiest to develop. It generally has a single path with a fixed number of steps. The titration experiment is an example of a reasonably linear simulation.

Cyclic contexts, in which events are repeated (perhaps slightly different each time), are fairly easy to simulate in the same way.

Contexts with complex sequences are the most difficult to simulate.

They must usually be simplified a great deal before programming. Complex sequences usually have multiple paths with variable steps. Since the paths and number of steps depend on the actions of the user, they may change at any time.

COMPLETION OF THE SIMULATION

Completion means the student has succeeded or failed in a particular run through the simulation.

In process simulations, this means the process has run to completion, and the student may choose to begin it again or not.

In physical, procedural, and situational simulations, completion usually means the student has followed either a successful path or one that has led to failure. In either case, it does not necessarily mean the student terminates the simulation. The student may choose (if the option is available) to do the simulation again immediately. If the student does not choose to do so, the simulation is either temporarily or permanently terminated.

In Radar and Ship handling simulations, the termination of each scenario can be controlled by instructor. He can also freeze the program at any time for evaluation or for any purposes.

TESTS

INTRODUCTION

There are two ways to incorporate computers in the testing process. They are as followed:

(1) To use the computer as an aid to construct the test.

(2) To use the computer to administer the test.

CONSTRUCTION OF TEST BY COMPUTER

To construct the test program by computer, there are two methods.

First method

(1) Questions must be permanently stored first.

(2) Then ask computer to choose in random order or to sort them in the order given by the instructor.

Second method

(1) Create the general format or template of the question together with the procedure.

(2) To generate, let us say, ten questions and ask the student for the answers, the computer will substitute different numbers for each new question.

There are many other methods of making test programs.

COMPUTERISED TEST ADMINISTRATION

The important note in this context is that although the entire test is automated with the computer taking the role of the instructor, human factor is also essential to determine and conduct the test.

The test is determined by the instructor as followed.

(1) The content of the test can, and must be tailored to suit each individual, or different tests can be constructed for each student.

(2) The student's answer together with questions, time to completion, response patterns or frequency of seeking help, etc., are stored in the computer memory. They provide useful information for the instructor to improve testing process.

Computer-managed learning **(CML)** and *computer-assisted training* are the two parts that combine to make up *computer-based training***(CBT)**. CML has been mentioned in Chapter I. It is a complex process for the computer to do well, and some of the facilities described here are only provided with CBT systems running on minicomputers and mainframes and not necessarily with all of them. It is by no means essential to do any of the course management on the computer, but it must be done somewhere if the training is to be methodical and complete. United States Merchant Marine Academy is one of the places where management of academic programs is done by computer.

Computer-managed Instruction **(CMI)** or *Computer-managed Learning* **(CML)** does not, of itself, require the trainers to use the computer to present the training material; this was done with a project called **CAMOL** (Computer Assisted Management of Learning). The computer can keep details of the courses and modules that each student has completed, whether it was lecture, audio-visual, practical, etc. It can record any marks and issue instructions to the student about the next modules to take and the sequence. It can also report to the supervisor on student progress.

There are several functions of CML that can be considered separately and, at the lowest level, virtually all CBT systems provide some, notably statistics, and the ability to direct the student to new material, depending on cumulative performance or a (series of) test(s). We shall consider the following:

- # Registering courses and students
- # Testing and record-keeping
- # Course maintenance
- # Reporting
- # Running a CML system

REGISTERING COURSES AND STUDENTS

Courses will probably consist of a number of modules, some of which are computer-based even if they incorporate other media, and some are in other media altogether or consist of a practical experiment. It should be possible to establish a course structure defining the path that students will go through, depending on various criteria at each critical point, and this will have been done at an early stage of the design process. Data on student performance can be accumulated during the module, but this is rather more difficult to plan and validate.

The course structure is then established and will consist of a student progress chart, of the type shown in Figure 2-25 which will be recorded in the computer. This indicates how that particular student will be assessed at each stage in a course. All students on a given course will normally have the same progress chart. It is then necessary to register the students for the course, so that records can be established of their progress and, possibly, for security purposes. If the company has a widely used terminal network, it may be quite inappropriate for anyone who can get into the system to be able to use the training programs.

Also, as trainers, one will wish to ensure that those who do sign on to one's courses are properly equipped with the essential supporting material.

TESTING AND RECORD-KEEPING

If CML is used, the important information should be stored as part of the student record and accumulated as he progresses through the course.

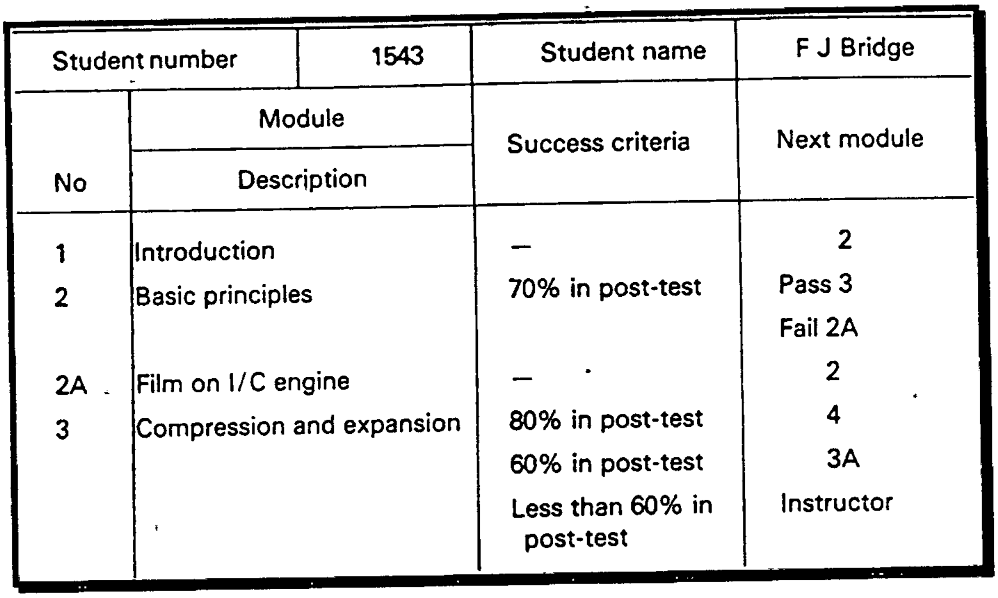
It is not essential or necessarily desirable for all, or even most, of the testing and assessment to be carried out by the computer. . It is quite possible to administer tests or assess a piece of practical work and then enter the marks for these into the computer. The marking can be done using special marked documents.

Whenever one discusses keeping any type of personal records on the computer, the problem of privacy can become an issue.**(Lecture given by Capt. Mc. Mullen, USMMA).** In this case, if properly handled, there should normally be no problems. Handling the records properly involves making it quite clear what data are being collected and why, and being willing to let the student see his own records if he wishes.

The fact that handwritten comments on the individual in his personnel file may be far more derogatory and inaccurate than those on the computer. . However, the inaccessible, is not strictly relevant. There are also the legal requirements to be considered. In the UK, these are covered by the Data Protection Act.

So, what data will be kept? It will depend on what you require but may include any or all of the following:

- Course name
- Student name and number
- Date started and date completed for each module
- Marks in tests
- All responses (these can take up a lot of disk storage, but are useful for validation).
- All frames passed through (these can take up a lot of disk storage, but are useful for validation).



Student number	1543	Student name	F J Bridge
	Module	Success criteria	Next module
No	Description		
1	Introduction	—	2
2	Basic principles	70% in post-test	Pass 3 Fail 2A
2A	Film on I/C engine	—	2
3	Compression and expansion	80% in post-test	4
		60% in post-test	3A
		Less than 60% in post-test	Instructor

Figure 2-25. An example of a student progress chart.

- Time spent on each module
- On remote terminal systems you may record the locations of the terminals used.

COURSE MAINTENANCE

It is often easier to maintain and amend CBT courses than conventional courses. Very few CBT courses use the computer alone. The changeability of the information is therefore one factor to consider when deciding the best medium for specific teaching material.

This is not to suggest that invalidated parts of courses should be put up in this way; it is rather that inadequacies came up by ongoing validation of student responses and comments should be corrected.

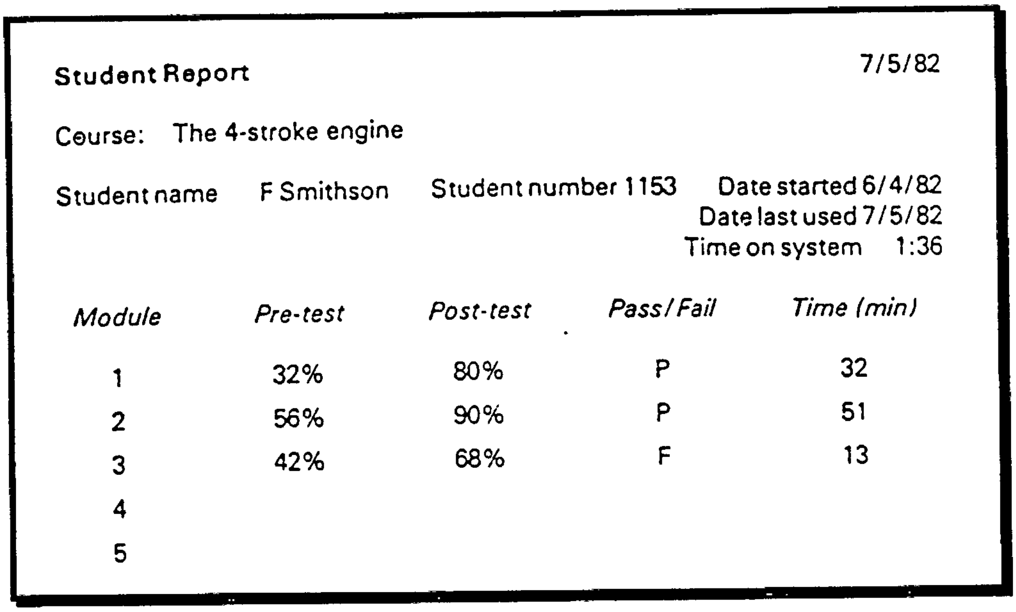
Other aspects of course maintenance involve adding, replacing, and deleting modules. The implications of altering the module structure in a course may be far-reaching, as it will alter the information recorded in the computer for the student progress chart (see Figure 2-25). The CML system should not allow anyone to add or delete modules unless one has to restructure the student progress chart. It is also quite possible to use a module or a group of modules in more than one course. If one starts altering these, he has to be careful or he may cause confusion even to the most sophisticated CML system.

REPORTING

The instructors need reports on the students' progress while authors need information about the courses. The training manager may need additional reports on the utilization of the system, on the times students take for courses and where they are located.

STUDENT REPORTS

The information that instructors and management can obtain about each student on a course will depend on the system, but they generally wish to know how far the students have progressed, the results of the tests, the time spent at the terminal and, most important, which modules they have mastered, i.e. if an acceptable level has been achieved. A report might look like the one in Figure 2-26.



Student Report 7/5/82

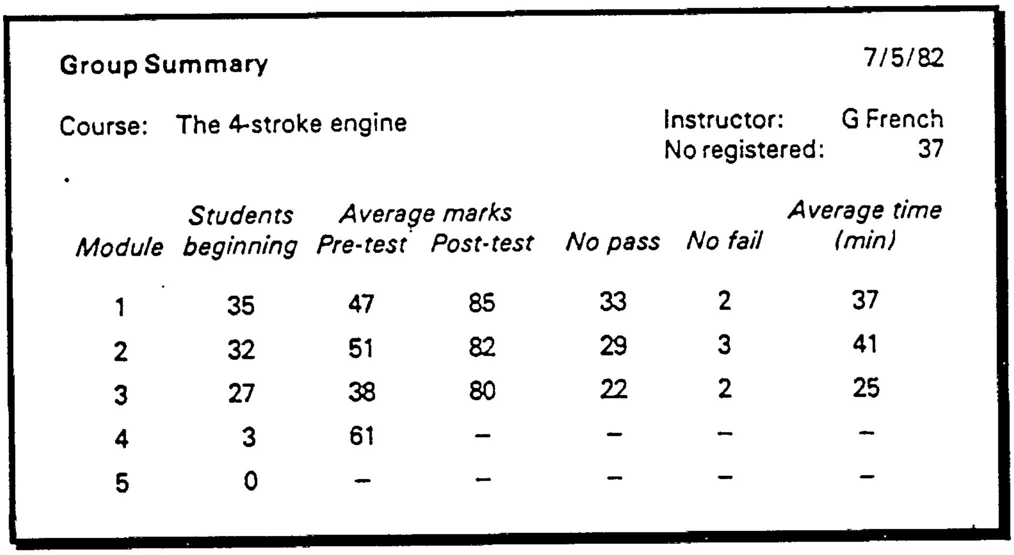
Course: The 4-stroke engine

Student name F Smithson Student number 1153 Date started 6/4/82
Date last used 7/5/82
Time on system 1:36

Module	Pre-test	Post-test	Pass/Fail	Time (min)
1	32%	80%	P	32
2	56%	90%	P	51
3	42%	68%	F	13
4				
5				

Figure 2- 26. A student report indicating the student's progress.*

It is also possible to obtain a student summary report, so that an instructor can have an overall view of how the students under him are progressing on a particular course (see Figure 2-27).



Group Summary 7/5/82

Course: The 4-stroke engine Instructor: G French
No registered: 37

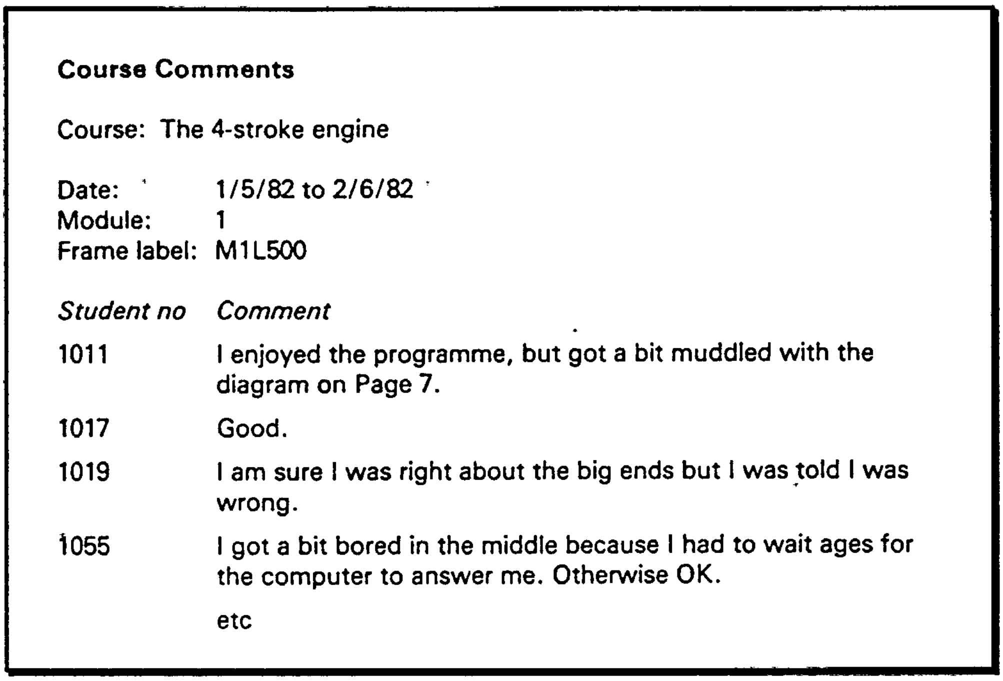
	Students	Average marks				Average time
Module	beginning	Pre-test	Post-test	No pass	No fail	(min)
1	35	47	85	33	2	37
2	32	51	82	29	3	41
3	27	38	80	22	2	25
4	3	61	–	–	–	–
5	0	–	–	–	–	–

Figure 2-27. A group summary for a particular course.*

COURSE REPORTS

A huge amount of information about student responses, the values of counters and the paths that students take through the course can be recorded on disk files. It is usually only practical to accumulate this information soon after release of the course, as the last stage of validation. It is a tedious job for staff to keep records for all courses. It may be important to make good use of this information for the first 20 or 50 students who take a course and then cease to collect it.

Another report that you can produce will print student comments. Depending on the CML system and how you use it, it may be possible for students to make comments at any stage in the course, at specific points, or only at the end. This facility does not use up a large amount of disk space, so it is sensible to keep it going permanently and print comments every month or so when the course is in general use (see Figure 2-28).



Course Comments

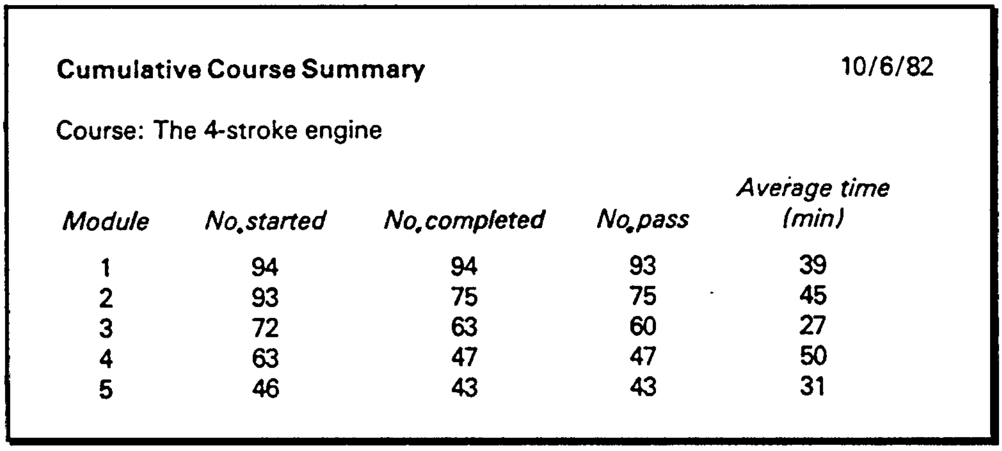
Course: The 4-stroke engine

Date: 1/5/82 to 2/6/82
Module: 1
Frame label: M1L500

Student no	*Comment*
1011	I enjoyed the programme, but got a bit muddled with the diagram on Page 7.
1017	Good.
1019	I am sure I was right about the big ends but I was told I was wrong.
1055	I got a bit bored in the middle because I had to wait ages for the computer to answer me. Otherwise OK.
	etc

Figure 2-28. Comments on the course from the students .*

Other reports that will be of use to the course authors and to the training manager will give summaries of the level of use of the courses. An example is given in Figure 2-29.

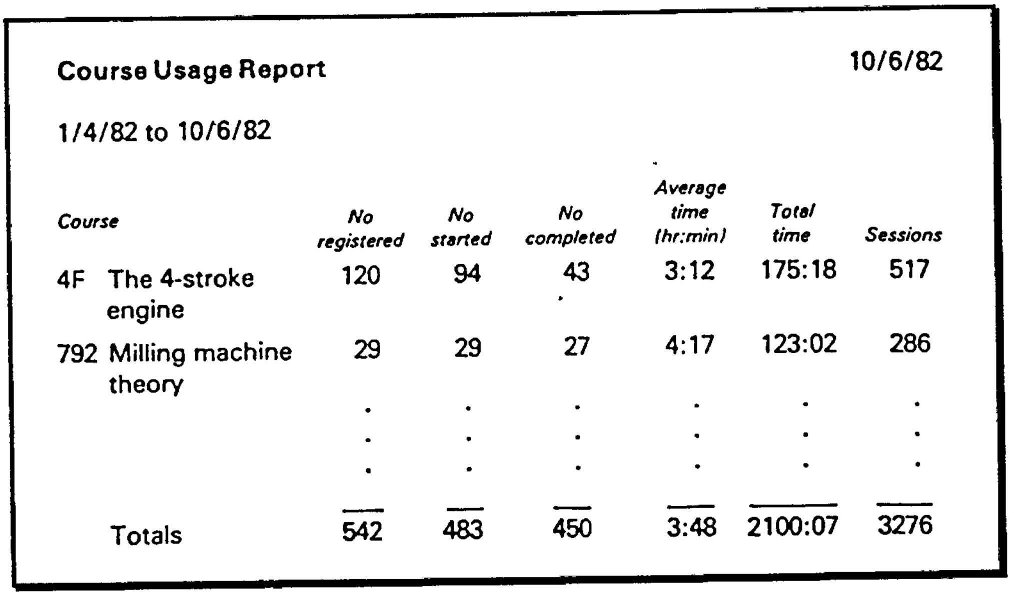


Cumulative Course Summary 10/6/82

Course: The 4-stroke engine

Module	*No.started*	*No.completed*	*No.pass*	*Average time (min)*
1	94	94	93	39
2	93	75	75	45
3	72	63	60	27
4	63	47	47	50
5	46	43	43	31

Figure 2-29. A cumulative course summary showing the use of a specific course.*



Course Usage Report 10/6/82

1/4/82 to 10/6/82

Course	No registered	No started	No completed	Average time (hr:min)	Total time	Sessions
4F The 4-stroke engine	120	94	43	3:12	175:18	517
792 Milling machine theory	29	29	27	4:17	123:02	286
	.	.	.	.	.	.
	.	.	.	.	.	.
	.	.	.	.	.	.
Totals	542	483	450	3:48	2100:07	3276

Figure 2-30. A course usage report showing all courses are being used.*

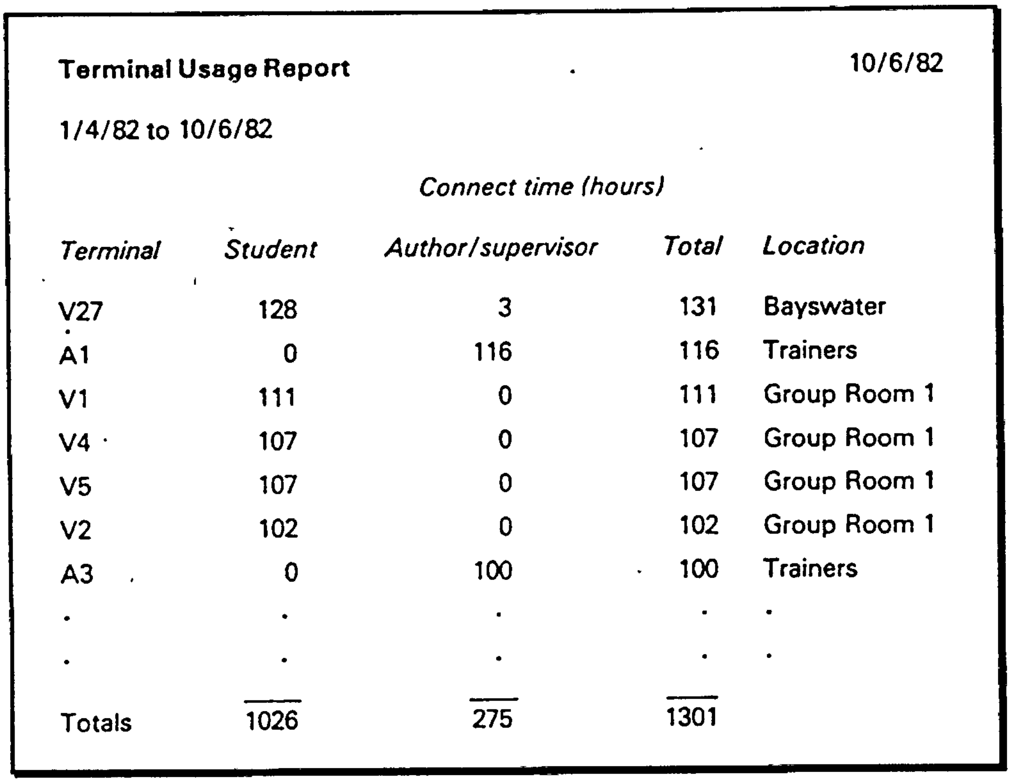
MANAGEMENT REPORTS

Training management will wish to know how much the various courses are being used, how long students take to complete the courses and the utilization of terminals.(See figure 2-30 and 2-31).

RUNNING A CML SYSTEM

Certain aspects of the administration of a system with sophisticated CML features require specialist skills which at least one member of the training department should acquire, with the data processing department providing back-up.

The CML administrator will be responsible for setting up the course modules and the linkages from the student progress chart. He will assign numbers to students at the request of instructors. He will produce standard reports and advise training personnel of other reports that could be produced, although the data processing department may amend the programs that print the reports, once the layout has been agreed. He will also be responsible for monitoring the performance of the system and the use of computer resources.



Terminal Usage Report 10/6/82

1/4/82 to 10/6/82

		Connect time (hours)		
Terminal	*Student*	*Author/supervisor*	*Total*	*Location*
V27	128	3	131	Bayswater
A1	0	116	116	Trainers
V1	111	0	111	Group Room 1
V4	107	0	107	Group Room 1
V5	107	0	107	Group Room 1
V2	102	0	102	Group Room 1
A3	0	100	100	Trainers
.	.	.	.	.
.	.	.	.	.
Totals	1026	275	1301	

Figure 2-31. A terminal usage report showing which terminal are being used.*

The CML administrator may receive various specialized reports showing, for example, response times for students, i.e. the time the student has to wait after completing his entry before the computer responds. A wait of more than a second or two is not usually acceptable unless something is happening, or, the student is warned. Other reports might inform the administrator or the amount of disk space he has allocated to him and how full it is, and the amount of central processor time the courses are taking up. This would indicate the training load on the system. He would also get an error report if anything went wrong, whether the system recovered automatically or not.

Whether the CML administrator spends his whole time running the training system depends on the level of usage; if a high level of sophistication in the computer management of the learning is being paid for, it is not going to be cheap. One should therefore plan to use it widely if he gets it.

[#1] **Mr. Burrhus Frederic Skinner** is a famous psychologist who, from his own research, described *human behavior* through science. His work has led to the *programmed learning and teaching machines.*

[#2] The **OOW simulator** is a PC-based simulator produced by **PC Maritime Ltd,** UK. They also produced marine softwares for training, operational navigation and weather information. The company has won two UK government awards for research and development, and three of the company's navigation programs have won design awards.

* Source:**(Dean & Whitlock, 1989.)**

CHAPTER III.

DEVELOPING AND DESIGNING LESSONS

A BRIEF OVERVIEW OF METHODOLOGIES

The methods mentioned here are only for developing computer based lessons, nothing concerned with other purposes such as teaching by programming or administering classrooms.

There are two models among computer based instructions, namely; **ISD** *(Instructional System Developments)* and **TICCIT** *(Time-shared Interactive Computer-controlled Instructional Television System).*

ISD MODEL

Generally, ISD model specifies only behavioural objectives, analyses learning tasks and activities for certain levels of student performance. It is more effective when done in perspective of entire curriculum rather than just the individual lessons. The **ISD** approach, therefore, has little influence on the peculiar nature of computer displays and keyboards, the nature of immediate feedback available on computer and on programming the computer. Its instructional methodology involves only *presentation practice*, and *testing* like the methodologies such as *tutorials and drills* mentioned in chapter II.

The **ISD** model was not originally developed for the purpose of producing computerised instructional materials. It was developed for the purpose of producing instructional materials in the form of texts, films, video and actual field experience.

In summary, although ISD provides a comprehensive approach to curriculum design for standard media, it lacks sufficient detail at the level of individual lesson development. It provides little guidance regarding the more creative uses of the computer such as simulation and games.

TICCIT MODEL

The instructional philosophy of this model has been discussed in Chapter I. There are two primary considerations in the original TICCIT project, and they are:

(a) The instruction should be expository with direct explication to the student of various facts, rules, concepts and principles. This presentation is followed by examples of above mentioned facts, rules, concepts and principles. Then practice follows.

(b) The second consideration is that the sequence of instruction must be learner controlled. The student should have access to control not only over that part of curriculum, but also within any part of that curriculum including the order of presentation such as initial explanations, examples, exercises, tests, and even difficult level of material.

Owing to its learner controlled instruction paradigm, there are two drawbacks, and they are as follows:

(a) It is limited to adult populations who are expected to make reasonable decisions about what to do next within the learner controlled environment. That may not be a problem in maritime education, because students in maritime institutes are being adults.

(b) It primarily restricts instruction to a strictly expository approach that precludes simulations, games, and other creative use of computer.

Nowadays, there is on the market, a new authoring language called **TAL** *(TICCIT Authoring Language)* that will allow programming of applications, like simulations. However, it is impossible to implement this on a computer that is not equipped with the authoring system.

This kind of authoring system can be found also on the **Apple Macintosh** computers, called Hypercard program. Unlike PC computers, MacIntosh has a very good power of producing high quality colour resolutions. **Dennis J. Lye(1991)** mentioned that the Hyper card program is not difficult because it builds upon the graphic user interface of the MacIntosh computer.

Coming back to the above mentioned two models, the severe short coming is the degree to which they inhibit creativity. The computer in our belief, is a new educational tool capable of delivering instruction in new and interesting ways. Simulations and games have, no doubt, great potential for improving instruction in all subject areas.

By knowing the shortcomings of the above famous models, one should consider the following factors in creating CBI programs.

(1) The lessons should not be dry and tedious.

(2) They should incorporate interesting and sophisticated feedback, branching, graphic demonstration and analogy.

(3) Good Interaction between computer and student.

SYSTEMATIC PROCEDURE

To achieve above mentioned goals, systematic designing of the lesson is utmost important. **Alessi &**

Trollip(1985) laid down eight steps for developing lessons in this context, and they are summarised as followed.

(1) Define purpose.

(2) Collect resource material.

(3) Generate ideas for the lesson.

(4) Organise ideas for the lesson.

(5) Produce lesson displays on paper.

(6) Flowchart the lesson.

(7) Program the lesson.

(8) Evaluate the quality and effectiveness of the lesson.

STEP 1: DEFINE YOUR PURPOSE

(I) State the general goal for the subject area. State what part of the subject area you intend to teach in a single lesson.

(II) Study the characteristics of intended students.

(III) State a terminal goal for the lesson.

(IV) Estimate how much time it will take and how difficult it will be to learn to reach your terminal goal.

STEP 2: COLLECT RESOURCE MATERIALS

(I) Produce a list of resource materials you intend to collect for the subject matter, for instructional development, and for the operation and programming of your computer. Be sure to include knowledgeable people in each category. If you do not know of many resources, go to the resource people for help in generating a list.

(II) Collect textbooks, reference books, manuals, films, videotapes, and other materials reflecting the content of the subject and its organization. Be especially watchful for organizational summaries, indices, tables of contents, charts, and graphs.

(III) Collect other instructional design texts. Use their bibliographies to locate other useful resources. Collect paper and various graphic arts materials such as graph paper, clear plastic, and a ruler.

(IV) Have your computer available, and collect manuals and reference guides for it and the programming language you intend to use.

STEP 3: GENERATE IDEAS FOR THE LESSON

(I) Gather three to five people around a small table.

(II) Provide each person with the instructional goals, resource materials, paper, and pens.

(III) Produce as many content ideas as possible without being judgmental.

(IV) Stop when few new ideas are being generated.

(V) Give each person a list of instructional methodologies and factors.

(VI) Repeat the process to generate ideas about how to teach the things listed in the first brainstorming session.

(VII) Stop when few new ideas are being generated and a few ideas have been generated corresponding to each idea on the first list.

(VIII)Save the two lists of ideas for the next step.

STEP 4: ORGANIZE YOUR IDEAS FOR THE LESSON

(I) Eliminate ideas based on student characteristics, subject matter, time, and delivery system limitations.

(II) Analyse the ideas that remain to fill out the details and sequence of the lesson.

(III) Choose a methodology or combination of methodologies.

(IV) Make decisions about treatment of all the instructional factors for the methodologies and ideas they will be applied to in the lesson.

(V) Arrange the results of the above four activities in a preliminary lesson sequence.

STEP 5: DISPLAY PRODUCTION

(I) Write and revise primary text, which includes information, questions, and feedback. Text can also be scanned by a scanner if available and.saved for later use.

(II) Write and revise secondary text, such as directions, hints, and end-of-lesson messages.

(III) Produce storyboards, in which you rewrite all of the materials from the previous two substeps to fit on your computer display.

(IV) Check the fit of overlapping displays, such as questions and feedback or information and directions, to make sure that nothing overlaps and that displays are not too crowded.

(V) Draw and ŕevise graphics displays and plan other output, which includes pictures, graphs, cartoons, sounds, music, and voice. Graphics should be done, if possible, in the actual size they will appear on the computer display. Pictures can be scanned by a scanner if available, and saved as files for later use.

(VI) Check graphics and simultaneous text for fit. Pictures and their descriptions or directions, for instance, should be co-ordinated to eliminate overlap or crowdedness, as in Substep 4 above.

(VII) Assemble the storyboards on a bulletinboard, blackboard, or large table so they can all be seen at one time.

(VIII) Review the storyboard assembly, looking for errors, checking for completeness and getting, for the first time, the whole picture of the lesson.

(IX) Have others review the storyboard assembly. Write down all comments that reviewers make.

(X) Mark revisions. Fix real problems, but not little details. There will be lots more changing to come in the next two steps.

STEP 6: FLOWCHARTING

(i) Know and use the flowchart symbols.

(11) Start the flow charting process by a simple flowchart.

(iii) Increase the level of complexity until the details are specified.

(iv) Use more and proper subroutines when possible.

(v) Dry-run the program (running without computer)

(vi) Revising and dry-running repeatedly until error free.

STEP 7: PROGRAMMING THE LESSON

BEFORE YOU START

Choose a programming language, enhancement, or authoring system that meets your instructional needs.

ERROR PREVENTION:

(I) Make a detailed (level three) flowchart of your lesson.

(II) Dry-run the flowchart before programming.

(III) Refer to the flowchart while programming.

(IV) Program subroutines first.

PROGRAMMING

(I) Write the first draft of the program off-line.

(II) Whenever possible use code that you have written before.

(III) Whenever possible amend existing code rather than write new code.

(IV) Whenever possible, modularise code for efficiency.

(V) Use tracing paper for graphics, placing it over the screen or graphics tablet, if any, when creating pictures.

(V) Use tracing paper for graphics, placing it over the screen or graphics tablet, if any, when creating pictures.

.(VI) Ensure that the informational detail of each display is in harmony with the instructional objectives.

(VII) Document your code thoroughly. Make changes to the documentation whenever you change the code.

(VIII) Always make your program work before making it efficient.

ERROR DETECTION

(I) Find and eliminate known bugs first.

(II) Use a systematic approach to error detection.

(III) Plan your debugging with data whose consequences can he easily determined.

(IV) Use a set of responses that will give predicted results.

(V) On each display, press all the keys. Only expected ones should have any effect.

(VI) Whenever a response is expected of a student, typing in unexpected answers, non-sense, and nothing at all. See how the program responds.

(VII) Use debugging tools provided for the computer.

(VIII) Print some identifying numbers on displays to facilitate reference.

(IX) Eliminate one bug at a time.

(X) Save existing code until you are certain you do not need it.

STEP 8: EVALUATION OF THE LESSON

QUALITY REVIEW

(I) Use the checklist.

(II) Check the language and grammar.

(III) Check the surface features.

(IV) Check questions and menus.

(V) Check all invisible functions.

(VI) check the subject matter content.

(VII) Check the off-line material.

(VIII) Revise the lesson.

(IX) Apply the same quality-review procedure to all revisions.

PILOT TESTING

(I) Enlist about three helpers.

(II) Explain pilot-testing procedures.

(III) Find out how much they know about the subject matter.

(IV) Observe them go through the lesson.

(V) Interview them afterwards.

(VI) Revise the lesson.

(VII) Pilot-test all revised lessons.

VALIDATION

(I) Use the lesson in the setting for which it was designed.

(II) Use the lesson with students for which it was designed.

(III) Evaluate how the students perform in the setting for which you are preparing them.

(IV) Obtain as much performance data as you can from different sources.

(V) Obtain data on student achievement attributable to the lesson.

(VI) Obtain data On student attitudes toward the lesson.

DEVELOPING TUTORIAL PROGRAM ON LIFEBOAT EQUIPMENTS

STEP 1: OBJECTIVES

« Student will be able to recognise various lifeboat equipments that are required to be carried on a cargo vessel.

« student will be able to explain how the equipments are used on a lifeboat.

STEP 2: RESOURCE MATERIALS USED

They can be broadly broken down as below, and each item will be broken down in detail later.

Material	Purpose	Source	Remark
Text books and other maritime magazines.	For display of text and graphic	Library	Photocopies of some pages are needed

Material	Purpose	Source	Remark
Computer Hardware	For programming the lesson and for student's learning	In-house, lease or purchase	It must have enough memory on board
Computer software	For manipulating texts and graphics and programming	Shareware-libraries, software-stores	Update the product information and prices.

DETAIL OF MATERIALS USED

Material	Detail	Remark
Books	(1) **SOLAS, 74** with 1983 amendments, published by IMO. (2) **Survival At Sea.**[#1] (3) **Safety At Sea magazine.** (4) **Admiralty Manual of Seamanship, Vol.1**[#2]	Look for books with good quality pictures.

Material	Detail	Remark
Computer Hardware	# IBM/PC compatible. # Processor: 80386SX # Speed: 25 Mhz. # RAM[1]: 2 MB expandable to 16MB. # Drives: 3.5" high density floppy and 40 MB hard disk. # Display: SVGA monitor[2] with 1 MB-VGA[3] card. # Input Device: Mouse & Keyboard. # **Hand Scanner** with *OCR* & drawing software.	For convenience, it is better to have your own computer.

Material	Detail	Remark
Software	(1) **GW-BASIC** Programming Language. (2) **Quick Basic** Programming Language. Ver.4.5	1) To record student's name, ID No., Class, etc. (2) To record starting and stopping time of learning the lesson. (3) To display (1) and (2) in the Score Report. (4) For retrieving multiple choice questions and various feedback, in random order. (5) To store and randomize questions and the correct answers for them and decide whether student chooses correct ones. (6) To calculate scores according to student's correct answers. (7) To create Score Report which can be printed.

Material	Detail	Remark
Software		(8) To compile the program files written in **BASIC**[4] for student's registration and test into execution files by Quick Basic(with extension name of **EXE**) to work with PC Storyboard.
	(3) **PC Storyboard-Live Ver.1.0**	(1) To draw, fill colours, and edit figures and pictures that are either drawn in this program or loaded from scanned pictures(scanned by the Hand scanner mentioned previously). (2) To write texts in various attributes(e.g. Outline, italic, bold, shadow, etc.)

Material	Detail	Remark
Software		(4) To create a Menu and setup student's option in which student can have a choice: ⇒ to go back to Menu at any time during learning, ⇒ to go out to DOS prompt via Menu, ⇒ to jump to any module via Menu, or to repeat the same module, and ⇒ to carry out test after learning all modules.

STEP 3: GENERATING IDEAS

BRAINSTORMING

The first thing we have to do is brain-storming. It can be done alone, but preferably with a group of people. Write down every idea coming up from anyone, on a piece of paper. After being determined that no more idea would come out, then this is the time to stop brainstorming. The process of selecting and refining the best ideas is done in a later stage.

Brainstorming should be in two sections **(Alessi & Trollip, 1985)**. First, it is to generate ideas about *what information to teach,* using the previously defined goals and the subject matter resource materials from steps 1 and 2.

Second, to generate ideas about *how to teach* the information, using the lists of instructional methodologies and their respective instructional factors.

SEEKING IDEAS

(1) From relevant text books: - Make a note on the following facts while studying the subject matter in the text books, ·for example;

(a) How is the quality of text and illustrations as to readability?

(b) Are the pictures realistic?

(c) Can the pictures focus the student's eyes on what they want them to watch?

(d) Do the texts and illustrations describe exactly what the SOLAS said?

(2) From similar programs on the market: - There are similar programs that are produced commercially such as programs from **Videotel Marine International Ltd**. Their programs like "Buoyage Systems", "Principles of Radar" and "International Code of Signals" are the examples. From them, I realized that the lesson should be used either independently or as a supplement to the existing text books, and that it should be a tool for the student to build up his retention of the lesson.

(3) From the experienced personnel: - Advice from resident and visiting lecturers is valuable to generate ideas. **Capt. Steven J. Cross**(previous advisor to the Academic Dean of WMU) and **Capt. Samar J. Singh**(visiting professor and Principal

Lecturer, Hong Kong Polytechnic) advised me to work on this program.

STEP 4: ORGANIZING IDEAS

As a prerequisite, the following studies should have been done for an effective CBI program.

- Basic principles of CBI and screen designing techniques
- A considerable knowledge on computer hardware.
- Fundamental knowledge in Programming in BASIC LANGUAGE.
- Hand-on experience on software like PC Storyboard and/or a computer-drawing program.

After generating ideas, I could organize my ideas to start my lesson program on Lifeboat Equipments

To achieve an efficient program, the following factors have to be considered.

(1) Choosing the right student's population.

(2) Relevance of subject matter, available computer resources and objectives.

(3) Amount of time to teach.

(4) Restrictions and degree of ease of use of the lesson.

I chose the target population for my program as those who have been at sea and familiar with lifeboat fittings and basic seamanship terms, because I could be relieved from teaching lessons like *Lifeboat construction and fittings*.

I could arrange to put my lesson program on a 3.5" high density floppy diskette(1.44 mega bytes). There is only one Menu in my lesson and thus makes it less complex. There are some restrictions that the student is unable to choose a specific equipment item in the lesson, although he can choose a particular module. However, this can be done by using a special feature in the software called "Story Editor". Due to the fact that it is only an example and the limited diskette capacity, it was omitted.

As **Gagne & Briggs(1979)** mentioned that task analysis which is an integral part of the ISD and similar instructional design models, should be done on your ideas to get an effective lesson sequence. The purpose of task analysis is to break down complex skills into component skills. A good teaching sequence should first teach skills that only require the student to use and combine already acquired skills. The teaching sequence should proceed to combine these new skills to teach the student more complex ones, continuing to do this until the finally learns the skills that are the terminal objects of the lesson.

Following this way, I divided my lesson into two modules, namely; **Module 1: Description of Equipments** and **Module 2: Use of Equipments**. They are shown in the Menu in a sequential order, i.e. Module 1 is placed above Module 2, as instructions in Module 1 are much easier and simpler than those in Module 2. In each module, items are also arranged so that simpler and less-textual items will be learned before the complex ones.

STEP 5: PRODUCING LESSON DISPLAYS

REQUIRED TOOLS

The following stationery and drawing tools are needed.

□ Paper, both lined and unlined, pencils, coloured pencils, rulers and other art materials.

□ Photo copies of pictures and illustrations from text books as mentioned previously.

- □ Flowcharting Template or any computer software for drawing flowcharts.
- □ If and when available, a person who has had experience in designing instruction for computer presentation and a person who has had experience in BASIC programming.

Nowadays, fast developing computer hardware and software make life become easier, hence you may not need any drawing tools and papers.

PRODUCING GRAPHICS

The following steps have been taken while producing graphics for the instruction.

(1) SCANNING PICTURES FROM TEXTBOOKS

The hand scanner *(Logitech Scanman 256)* that I have with me includes two items of software, namely; **"Foto Touch"** and **"Perceive Personal"**. "Foto Touch" is used for scanning graphics of any kind of not larger than A4-size. "Perceive Personal" is for manipulating texts scanned by the scanner. The latter is known as **OCR** (Optical Character Recognition) software. Both software are **Windows™** applications. Figure 3-1 shows the picture of the scanner.

Figure . 3-1. Hand Scanner(Logitech Scanman 256).

Scanning of both graphic and text needs practice. The scanner has 256 grey scale switch by which I can have almost the same quality of photo-finished pictures on my computer, provided the program running on the computer must support grey scale display. Unfortunately, PC Storyboard that I have is neither a Windows™ application nor does it support grey scale graphics. Therefore, when scanned pictures are loaded into PC Storyboard, they are distorted. However, this was overcome by careful editing and touch-up of the pictures with its *editing tool*. Any sketches, drawings, figures and tables can also be scanned as *Line-art*, saved as **PCX**-format and can be loaded in the Storyboard without distortion. Figure 3-2 is the picture of a **sea anchor** scanned by the scanner and saved as PCX-format.

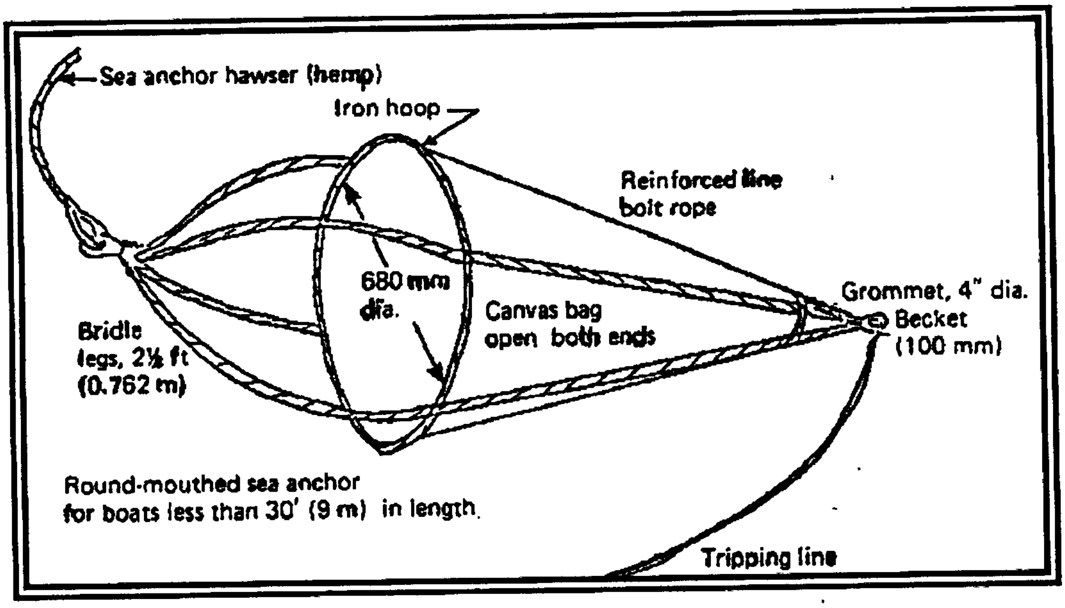

Sea anchor hawser (hemp)
Iron hoop
Reinforced line bolt rope
680 mm dia.
Canvas bag open both ends
Grommet, 4" dia.
Becket (100 mm)
Bridle legs, 2½ ft (0.762 m)
Round-mouthed sea anchor for boats less than 30' (9 m) in length.
Tripping line

Figure 3-2. A sketch of sea anchor scanned by the Hand scanner(saved in PCX-format). #1

The finished pictures, after editing and colouring in the PC Storyboard, were saved as PIC format. It is a special file format created by PC Storyboard and is recognized by it. The picture files in **PIC** format were named after the item of equipment, e.g. *COMPASS.PIC, CRUTCH.PIC, HANDFLA1.PIC(Handflare)*, etc. They were then saved in the hard disk(they can also be saved in a floppy disk).

(2) GRAPHICS DRAWN ON COMPUTER

Any sketch, line drawing, flowcharts, etc., can be drawn within it or in any other drawing tool (e.g. *Draw Perfect, Microsoft Draw, PC Paintbrush, Corel draw,* etc.). Many drawing software have a feature of saving a file in many formats that are supported by Windows 3.0(e.g. **BMP**, **PCX**, **TIFF**, **GIF**, etc.).

PRODUCING TEXT

Text required for presentation can be created by two ways.

(1) PRE-FORMATTED TEXT

Required text can be typed and saved in any word processor, as **ASCII**[5] file. There was a Menu option called *"Get Text"* in PC Storyboard and by this, I could load those text files into any picture file which has been loaded on the screen. Although the whole block of text after loading, could be moved anywhere on the screen, it is impossible to format the text and paragraph in such a manner that we normally do in a word processing programs. Therefore pre-designing of screen layout should be done on paper first, to allocate the space for text. If the space for the text is not enough, you should divide them into two(at least) blocks of text and display one block at a time without removing or resizing the associated picture. (There is an advantage that the picture could be resized on the screen, to be bigger or smaller.)

(2) TYPE-IN TEXT

In PC Storyboard, different type, size, colour, and attribute of text is available. Although line spacing is adjustable, paragraph formatting can be done only by manual. Alignment of text can be set for left, centre or right justify. New style of fonts can be created by an option called *"Font Editor"*.

In any case, once the text is positioned and the text option is abandoned, all text typed in or loaded become graphic mode.

OVERLAYING

PC Storyboard has a feature to overlay any graphic on top of any other both in transparent mode or in opaque mode. In transparent mode, the background graphic can be seen through the second overlaid graphic. In opaque mode, the second graphic covers the first one that can not be seen through.

This technique has been used in my program in displaying "prompt" for the student option at the bottom of each picture frame. These prompts can also be typed in permanently on the picture frames while creating PIC-format files. The second graphic can be laid partly over the previous one either by defining co-ordinates of the second picture or by selecting(blocking) the area by "Select" command.

STEP 6: FLOWCHARTING

A small instructional program may not require flowcharting. However, flowcharting nowadays can be done easily on computer by using any drawing software, e.g.-*Draw*

Perfect, Harvard Graphic, Flow, etc. With a click of mouse, any ready made shape or diagram can be loaded on the screen, thus saving time and preventing boredom.

The lesson design should and can be indicated by flowchart and it should be in a simple form that experts call first level flowchart. It contains no branches, and no explicitly stated decisions. The first level-flowchart for my tutorial program is shown in figure 3-3.

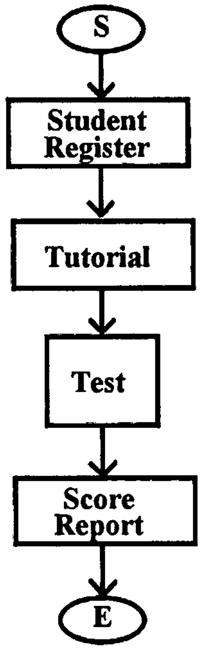


Figure 3-3. The first level flowchart for Lifeboat Equipment Tutorial Program.

The second level-flowchart contains a comprehensive overview of the structure of the lesson and includes most of the pedagogical attributes. Decision points are incorporated (see figure 3-4).

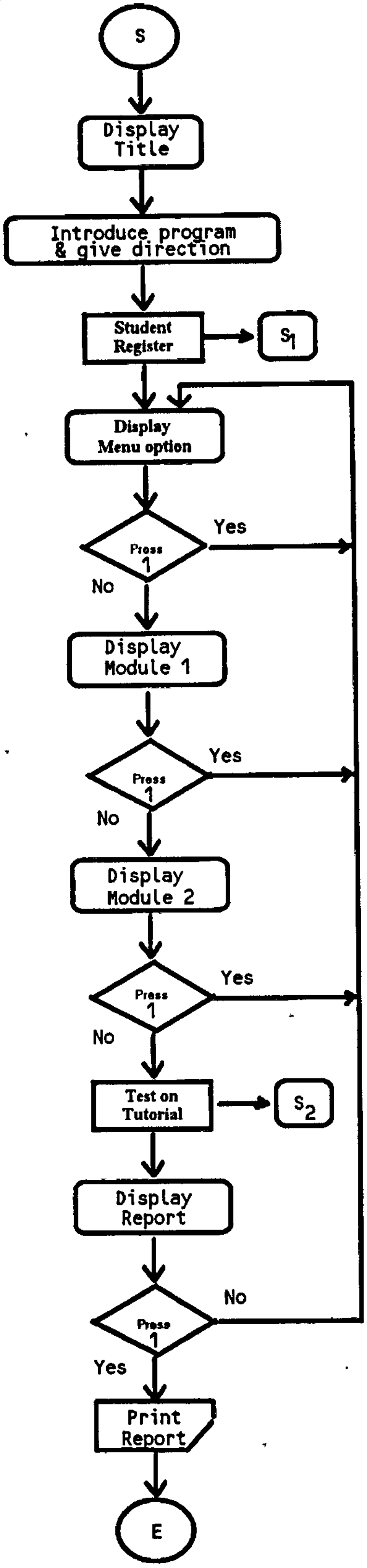


Figure 3-4. Second -level flowchart for Lifeboat Equipment Tutorial Program.

The third level flowchart provides the necessary detail for a programmer to produce the computer program for your lesson.

The main body of my program is run by PC Storyboard which is a bit far limited in complexity and branching. Programming involves only at the beginning whereby the student inputs his particulars for registration, and at the end where a test is carried out. The first program is called **Register.exe** and the other, **Test.exe**. The program listing of the above mentioned programs are attached at the back of this dissertation as Attachment (1) and (2).

STEP 7: PROGRAMMING THE LESSON

Although PC Storyboard is like an authoring language, it has restrictions and limits. It allows only two user's key-inputs in the learning process, i.e. it has two options that user can make, during presentation of a lesson item. For example, in my program, there is a prompt for the student to go ahead and a prompt to go back to Menu. There is no prompt to move back to the previous frame.

From the lecture received from **Capt. Samar J. Singh in 1992**, he taught us that there are two primary ways of instructing the computer what to do. The first is by means of a programming language, with or without the benefits of an enhancement. The second is by means of an authoring system. As mentioned before, a programming language, such as **BASIC** or **PASCAL**[6] requires that the programmer specify every detail of what the computer should do. Nothing can be left out; every aspect of answer judging has to be programmed, every variable or counter updated correctly, the sequencing of display taken care of, and the collection of data ensured.

If an enhancement to the programming language is available such as EnBASIC for BASIC, many of the more difficult routines are provided automatically. This makes the use of the programming language much easier. A language and enhancement have all the flexibility of the language, plus ease of implementing some procedures normally difficult to program. We like this combination: It is relatively inexpensive to acquire and quite easy to use.

An authoring system, such as **PASS**, already has most of the programming done, leaving the programmer the responsibility of providing the display details and some of the branching logic. All that has to be done when using an authoring system is to fill in the blanks provided by the system. If available, authoring systems provide an easy way to implement lessons, assuming the lessons meet the constraints of the system. Whatever technique you use to create your lesson, whether it be a programming language, programming language plus enhancement, or authoring system, there are some general procedures for improving the efficiency of the process.

Most of the recommendations, however, apply more to the use of programming languages, with or without enhancement, than they do to authoring systems. The discussion is divided into three sections: error prevention, programming, and error detection(**Hand out notes for MET-92 class from Capt. Samar J. Singh, 1992)**.

ERROR PREVENTION

It is very easy to accumulate errors into a program because the process is a lengthy and complicated one. Precautions can be taken, however, to minimise this error rate and to make it easier to find errors that are introduced.

REFER TO A FLOWCHART

A very useful procedure to follow in programming a lesson is to use an error-free flowchart as a guideline. This accomplishes three things. First, because the flowchart is structured in the same way as the lesson, the program will have a similar structure. Second, by carefully examining the flowchart it is easy to see which parts of the flowchart are used more than once.

These sections can then be coded in a way to facilitate this multiple use (usually by means of subroutines). Third, by checking off each part of the flowchart as it is programmed, you ensure that nothing is left out. A haphazard approach to programming will inevitably lead to omissions and errors. The most efficient procedure begins with a careful examination of the flowchart to locate all subroutines. Program these first, so that when you refer to them later they already exist and can be tested immediately. The same applies to text and graphic displays that are used repeatedly.

GENERAL MEASURES WHEN WRITING PROGRAMS

1. Using variable names that convey the function of the variable are a great help when the program has ended up with lengthy and complex. If you are using an interpreted language such as BASIC, these names can be easily altered in a word processor or editor with a single **"search and replace"** in the final version to conserve memory. In a compiled language the economy of using abbreviated variable names is meaningless. It is always nicer to read **DayOfWk%** instead of **d%**.

2. The same philosophy applies to filenames. Give them a version number in a REM statement in the program header as well as in the filename.

3. Keep **ON ERROR** routines in a **REM** statement until the programmes running satisfactorily.

4. Using **GOTO** is usually guaranteed to make your debugging task harder.

MEASURES AFTER THE PROGRAM HAS BEEN WRITTEN

· The program may run quite happily without error messages, but it may not perform the task you wanted. To determine that this is not the case you will need a series of test conditions. It is not unwise to prepare these when you write the program objectives, at the very beginning.

Alternatively, or what is more likely, is the program will not run as a piece of code, thereby giving the computer heartburn and causing it to generate error messages. The variables **ERR** and **ERL** can trap such errors and handle them gracefully.

Some people will find debugging from printed paper listing easier than from the screen. It will certainly be easier on the eye.

The following measures are handy when trying to debug such code:

1. Put a **STOP** statement at different stages of the program and list the relevant variables, you can then resume the flow of the program by using **CONT**.

2. Once the general area of code that is faulty has been determined, it is advisable to program the function keys to LIST the relevant part using the **KEY n,x$** statement.

3. To probe deeper the **TRON** and **TROFF** commands can be useful in checking the flow of control,

4. Often the error, 'No such variable' can arise due to the absence of a space as follows:

```
FOR LOOP%=START% TO END%STEP INCR%
```

5. If you have tracked a bug down to a given subroutine, it is sometimes useful to delete everything but that subroutine and the subroutine it calls, and then run it with the test conditions. Languages such as Pascal and some versions of BASIC permit the concept of PROCEDURES that can

employ local variables and also permit passing parameters. This reduces the possibility of global variables interfering.

STEP 8: EVALUATION OF LESSON

PILOT TESTING

Pilot testing is the process of having representatives of your target population use and test your lesson while you monitor their progress and performance. Although other people may have already provided you with feedback on the lesson, there is no substitute for having real students do the same. It is an essential step in the evaluation process, and it should never be omitted.

Pilot testing is a six-step process, which should be done in the order outlined below. The steps are:

- Select the helpers.
- Explain the procedure to them.
- Find out how much of the subject matter they know already.
- Observe them go through the lesson.
- Interview them afterwards.
- Revise the lesson.

Each of these steps is important if you want to obtain useful information from the process.

SELECT THE HELPERS

The best helpers are students much like those for whom the lesson is designed. Find at least three: One should be representative of the best students; one an average student; and one representative of the slowest of the students that will use the lesson. This spread of capabilities will enable you to test informally whether the lesson meets the needs of your entire target population. For my case, I asked four nautical students of WMU to volunteer; one from class of 93, and three from 92.

Naturally, because of the small numbers involved, one can only use the information gathered in the pilot test as very helpful guidance rather than as a definitive statement of fact. What this means is that if the selected students all report favourably about the lesson, it is likely to be successful in wider use, but not definitely. In this way you can devote all your attention to one and will not miss anything important.

EXPLANATION

Before the pilot students begin the lesson you should explain the purpose of what you are doing. You should say that the lesson is in the process of being developed, and that it is essential to test it before releasing it for general use. Ask the students to proceed through the lesson as though taking it for credit, and not to ask you for any assistance.

Encourage them to make notes about the lesson whenever they have a comment to make. This note-taking is facilitated if you can provide some easy means for the student to identify each display they want to comment on. This can be done either by having a display number appear on the screen (in the top right corner, perhaps), or by having a paper version of each display available on which notes may be written.

You should also explain that you will observe them at all times and will ask for a variety of information about the content and operation of the lesson at the end of the session. Encourage the students to be very critical. In my case, I received many critical comments from them on my program after the first presentation. After the second time, I received fewer critics but a lot of touch-up things.

DETERMINE PRIOR KNOWLEDGE

Before the students begin the lesson, ensure that you know what previous exposure they have had to the subject matter, and how much they know of it. You must then relate this to whether each is a good, average, or poor student.

This information is important when interpreting the data you collect during the rest of the session.

OBSERVATION

Throughout the session you should unobtrusively observe the student interacting with the lesson. It is most surprising how much information you can glean from just watching body language. You can tell immediately when a student is having difficulty understanding the material, or is confused by the directions on the screen.

You can tell when a student is unsure of what to do, or of what options to choose. You can also tell when the person is enjoying the experience, and when displays are appealing or boring. As the student progresses, take notes about the behaviour exhibited, noting particularly the type of behaviour and where in the lesson it occurred.

FINAL INTERVIEW

As students complete the lesson, you should interview them. The purpose of this is to discuss any comments either of you may have written down.

The difficult aspect of the interview is that if you are the lesson's designer or programmer, you may well be told many things that bruise your ego. It is vexing, for example, to have spent many hours (sometimes hundreds) producing what you regard as a masterpiece, only to have someone tell you that it is not very good.

When a situation like this occurs, when a student is very negative about one of your lessons, you should resist the temptation to explain why a display was designed in a particular way, or why you had structured the content the way you did. It is easy to become defensive about your lesson, but doing so will only reduce the effectiveness of the entire process by inhibiting the student from making further comments. If you keep in mind that any criticism you receive during the pilot-testing phase will ultimately benefit your lesson, it will be easier for you to handle these negative criticisms. The type of information you should seek in this interview relates both to the content of the lesson and to its operation. Ask whether the structure and logical flow of the subject matter were appropriate to the student, or whether another structure would have been easier to understand. Find out whether the amount of control the student had for moving within the lesson was sufficient, or whether greater freedom or constraint should be incorporated.

You should also seek information as to whether the lesson functioned satisfactorily and as expected. Ask whether the directions, aesthetics, language, and available assistance could be improved.

In addition to obtaining the students' reactions to the content and operation of the lesson, ask them how they felt about it. Was it enjoyable? Boring? Interesting? Useful? Find out whether they thought that their peers would benefit from the lesson. These affective responses often provide useful insights into how the lesson is really perceived.

REVISE THE LESSON

Once you have accumulated the data from each student, you should decide whether the lesson needs further revision. Sometimes this is a difficult decision to make, particularly if the reviews were contradictory. A good strategy to follow if this happens is to have a few more students take the lesson and solicit their views. If you decide to make major revisions to the lesson, you should repeat the pilot-testing process with the new version.

SUMMARY

Pilot testing is crucial for the production of quality lessons and should not be overlooked. Your decision should concern the amount of such testing and revision to undertake, not whether it should be done. Good developers always regard their lessons as being imperfect, and are constantly eager to improve them. It is important to realise, however, that at some point the extra effort you have to expend to make an improvement may not be time-effective or cost-effective. You should terminate the development process when the lesson accomplishes its purpose, not when you finally regard it as being perfect.

VALIDATION

The final stage in the evaluation process is the validation of the lesson, which is the process of checking how well the lesson works in a real instructional setting.

No matter how positively the students in the pilot-testing phase reviewed the lesson, it is important to put the lesson through this final scrutiny.

There are two main reasons for the importance of this phase. First, the real setting in which the lesson is to be used invariably is quite different from where it was tested. The computer may be in the middle of a busy classroom, causing a steady flow of diversions to the student taking the lesson, or, all the pre-requisite skills may not be present to the degree you had expected, or, supplementary materials may not be available. Second, although every attempt is made in the pilot test to use helpers who span the range of abilities of the target population, it is always doubtful whether they do.

Furthermore, data collected from only three or four students can never generalize completely to the entire population. The true test of a lesson occurs when it is exposed to a large number of students in their natural setting.

[#1] **Survival At Sea.** 1988. Fourth edition (revised). Written by CH Wright. Published by Brown, Son & Ferguson, Ltd., Glassgow.

[#2] **Admiralty Manual Of Seamanship,** Vol.1. (1979). New edition. Published by Ministry Of Defence, printed for Her Majesty's Stationery Office. UK.

[1] **RAM** : - Random Access Memory in the CPU of a computer.

[2] **SVGA monitor:** - Monitor that can display horizontally 1024 pixels and vertically 768 pixels on the screen if the video graphic adapter has the memory of 1 mega bytes.

[3] **VGA** :- Video Graphic Adapter which is to be fixed in one of the bus-slots in the CPU unit.

[4] **BASIC** :- Beginners' All-purpose Symbolic Instruction Code.

[TM] Windows is the Trademark of the Microsoft corp.

[5] **ASCII:** - American Standard Code for Information Interchange. It is mentioned in most of the dictionaries as follows.

ASCII:- American Standard Code for Information Interchange. Pronounced ass-key. A standard way of assigning numerical codes to characters and control codes.

6 **PASCAL** :- is a programming language and was named for Biaise Pascal, the seventh century French mathematician. It was developed by a Swiss computer scientist, Niklaus Wirth. It is widely used by people in Education and scientists.

CHAPTER IV

MICROCOMPUTERS and MEDIA

PART ONE

MICROCOMPUTERS IN MET

THE NATURE OF MARITIME EDUCATION

The skills that an experienced mariner is expected to possess may be broadly defined as follows:

1. Navigation skills
2. Ship handling skills
3. Interpersonal communication skills
4. Management skills
S. Decision-making skills
6. English Language skills

In addition to this the mariner is expected to possess a knowledge base whose compass is considerable, extending from the interpretation of meteorological data to familiarity with dangerous chemicals.

While it can safely be said that most maritime training institutions cater to the needs of the extensive knowledge base, the matter of inculcating the different skills outlined above is a different one, and may not be directly included in either the educational or the statutory examination system. Few, however, would deny the need for these skills in the modern ship master.

Such skills epitomise understanding while the knowledge base represents information transfer. It is in promoting understanding that the microprocessor and its descendants, microcomputers and simulators may be potentially available.

MICROCOMPUTERS AND SIMULATORS IN EDUCATION

There are two different ways in which the introduction of microprocessor technology into the teaching process can benefit the budding shipmaster.

Firstly, understanding of the reliability and limitations of microprocessor based devices such as *Automated Radar Plotting Aid*(**ARPA**) and *Electronic Sea Chart and Information Systems*(**ECDIS**) make their inevitable entry into the spectrum of shipboard bridge equipment.

Secondly, microprocessor technology in the form of microcomputers and simulators can have a stimulating role in developing the skills outlined above as well as in aiding the process of information transfer. The importance of microcomputers in the educational process is underlined by the *Computer Literacy Project* in the U.K. and the *"10,000 Microcomputers Plan"* in France.

Most general applications of computers are based on their role in:

1. utility processing,
2. computer assisted learning,
3. simulation,
4. computer-aided assessment.

1. UTILITY PROCESSING

This definition is intended to include the use of word processors, data base programs, and spread sheets. While the utilitarian aspect of such devices is well illustrated by the use of word processor within a networked system at the Dept. of English, Hong Kong polytechnic(**Handout given by Capt. Samar Singh, 1992**).

2. COMPUTER ASSISTED LEARNING

In general, we have been able to experiment with the use of computer assisted learning in two ways.

Firstly, microcomputers have been used for analysis where such analysis would otherwise have taken time out of proportion to its benefits. It involves the use of microcomputer graphics for **Collision Risk Analysis Simulation**(*Hong Kong Polytechnic*).

Secondly, microcomputers have been used to provide programs for self learning. This is best illustrated by the **Differential Vector Drill Trainer**(*Hong Kong Polytechnic*).

3. SIMULATION

In terms of adding understanding to the information transfer process - a necessary element in learning-simulation has much to commend it. No better rationale for the use of simulation has come my way, than the following excerpt adapted from **McCabe(1986)** that cites why one would want to include simulation in teaching:

- ♦ to integrate material from several subjects
- ♦ to promote student participation in learning
- ♦ to provide a risk free environment for experimentation

- to provide a broad view of the subject at the start of the course
- to ensure a broad view of the subject at the end of the course
- to implement theory
- to enable students to respond to a changing environment
- to enable students to develop professional competence
- to enable students to be able to do things, as well as to know of things
- to expose students to problem solving under confusing and ill defined circumstances
- to enable students to discover relationships between concepts instead of being told them
- to arouse student interest in the subject
- to allow important elements of the course to be based on the actual performance of skill

Capt. Samar Singh mentioned in his lecture that *"while the context of the above was the use of simulation and games in teaching and learning in a tertiary level multi-disciplinary institution much of it; is equally applicable to specially built simulators such as radar or shiphandling simulators. While specialist simulators are frequently found in maritime training establishments, they do involve heavy capital expenditure. The use of simulation in simpler modes is less onerous financially and is well documented, and in my experience, under-utilised in maritime education."*

4: COMPUTER AIDED ASSESSMENT

Assessment can be broadly classified into three areas in the maritime training context.

Firstly, one may wish to have an assessment of data recall to evaluate the degree of information transfer. Valuable work has been done in this field and home grown and professional programs for objective type testing are available. In the maritime education field, **Odeh & Facey(1988)** have outlined a microcomputer based system for international standardisation of examinations for **STCW**(*International Convention on Standards of Training, Certification and Watchkeeping*) certificates. However, analogous solutions for essay type questions are not as frequently found. One may question whether this method of assessment is always necessary. Simulator based assessment could be an alternative in many cases.

Secondly, there is the question of the assessment of motor skills. These obviously play an important role in maritime training, in spite of the lack of standardisation in shipboard equipment. In any event, this requires hands-on access to equipment that may often be driven by microprocessor based devices. Radar recording and playback systems, and devices for synthesising signals to all manner of radio navigation receivers are in regular use at the Hong Kong Polytechnic for operation and maintenance training.

Thirdly, assessment in problem solving abilities is possibly the most important if least tested element of the mariner's training. Specialist simulators play an important part in such assessment. The example for this is the **Telex Simulation System** at the Hong Kong Polytechnic. It is used for the assessment of analytical and problem solving capability during Radar Simulator based courses.

THE QUESTION OF SOFTWARE

Although there are hundreds of commercially available software in the market, there is no exactly fit-in software for the maritime education. Additionally, it is difficult to tailor the software so that all peripherals, such as plotter, other type of computers, display units, etc., can be utilised without any problem. In house development of the software may be the best solution. However, the lecturer/user, should not be allowed to do that kind of job. The institutes should have a policy in allocating staffing resources for development of the projects.

PROBLEMS OF IMPLEMENTATION

There are a number of difficulties in implementing computer aided learning over and above those that arise from the financial, hardware, and software resource limitations. In general, it is a time consuming medium better suited to modular courses or seminars with small groups and low staff student ratios. If maritime academics can define curriculum areas appropriate for computer aided learning then some of the problems of implementation can be avoided.

McCabe(1986) has summarised the problems of implementing simulation and gaming as follows:

1. Such methods do not fit well with neat and tidy one hour "lecture period" that are the units of departmental timetables.

2. The normal methods of assessment are largely limited to the recall of facts or rote learning. We do not precisely know how to quantify higher order processes such as judgement, creativity, and problem solving that simulation and gaming techniques promote.

There is of course no reason that microcomputers cannot be used to produce superior rote skills, although this would imply a dubious basis for the replacement of traditional methods of teaching and learning.

There is no hard evidence that simulation is effective partly because of the absence of reliable indicators for the qualities that simulation is said to enhance. **Garvey*** notes that while the area of skills development does not seem to be one where there has been scientific investigation, his observations have been sufficiently convincing to state that simulation produces an increase in demonstrated competence of the skills employed.

As seen in Capt. Singh's handout **Boyd**** has summarised the negative aspects of computer aided learning as follows:

1. Incomprehensible and unrepairable machines can lead to apathy.

2. Pervasive planning for and with machines encourages us to treat each other as things.

3. Performance-giving machines rob people of the needed opportunities to give performances to their pupils and friends.

4. There is an element of time-wasting addictiveness to computer aided education.

In similar vein, **Hoyle***** has noted the role of the computer as a solution in search of a problem.

CONCLUSIONS

Like most technological advances, computer aided education may be described as a device that merely changes the problem to which it is applied. A proper understanding of the nature of change can highlight the pitfalls and underline potential areas of application. The maritime educator plays a key role in defining the optimal mix of traditional and modern methods, within specified objectives of maritime training.

PART TWO

AN OVERVIEW OF COMPUTER DEVICES AND MEDIA USED IN CBI

The purpose of this part of the chapter is to find out the devices used for storing data and the medium used in the CBI applications.

As mentioned in the Introduction of this dissertation, I assumed that we have already learned basic principle of how computers work and what are their hardware and what are meant by software, Dos, etc. We are now going to look how the flow of the data (text, sound, graphic, etc.) in the CBI should go through different medium and devices effectively. We will also look into the application of Multi-media instructions.

BASIC MODEL

In a simple Computer based instruction, the human-computer interaction is by the display unit(here we call, *CRT-Cathode Ray Tube*) and the *keyboard*. The display units for the IBM PC computers are specially made CRT, called *monitors*. TV can be used for some computers like **Amiga**, **Atari**, **Archimedes**, and ***BBC*** as a display unit, because they include a digital-video signal converter in their CPUs. The hardware makers are now producing this kind of converter as add-on card that can be installed in the **IBM PC** machines, but it is still a bit costly. **Capt. Samar Singh** has demonstrated the use of a computer that can do multi-tasking, during his lecture.

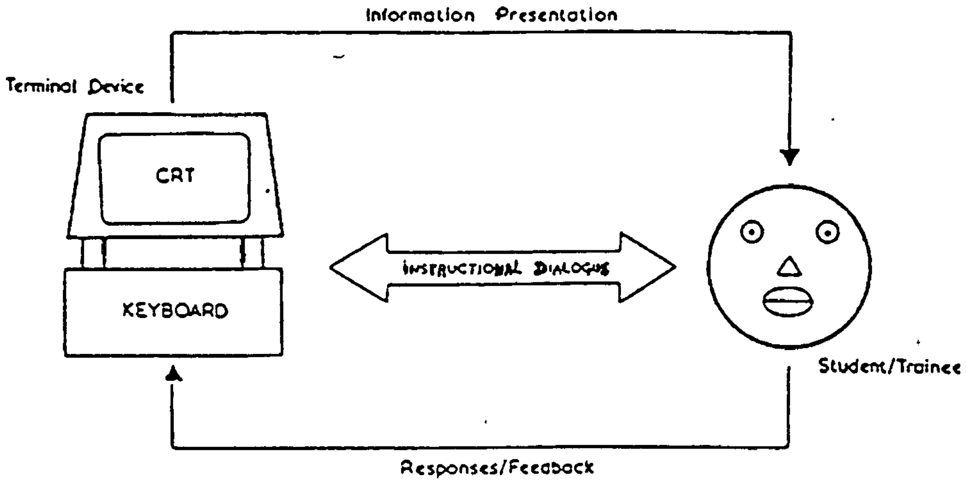


Figure 4-1. Basic model.

The keyboard/CRT model is illustrated in Figure 4-1. Essentially, a single cathode ray tube (CRT) display is used to present instructional material to a student or trainee. The advanced technology made the CRT to play an important role in the CBI applications in enhancing the information presented to the student.

From this model, we can integrate it into a Multi-media Computer assisted learning programs by connecting different peripheral devices.

GENERAL IDEA OF MULTI-MEDIA

As a lot of devices and interfaces are connected internally in the computer or externally, it is necessary to know how they are connected and designed to work together. It is usually called Resource pack and the instructions for this pack are contained in a *workbook*. This workbook is likely to contain sheets of instructions or pages of notes that are:

1. describing how to operate the multi-media workstation, and the course units,
2. explaining how to use
   - the resource materials and interface cards, for use in conjunction with various workstations (such as a *digitizer*, a *concept keyboard* or an *electronic book*), *video disc* and *CD-ROMs*, *compact discs* with read-only memory (containing text, pictures and sound), and
   - activity lists for the students to work through.

Within the resource pack there may also be special items of equipment. Among these might be included specific learning aids such as a micrometer, thermometer or microscope. Learning how to use these might be an integral part of the course of instruction.

Within a **multi-media CAL** environment it is important to realize the multiple roles played by the computer itself. Thus, the computer will be responsible for monitoring the student's learning progress and suggesting possible activities that might undertake. It may also control some of the other media (such as video, slides, or audio) during the learning/training session. Of course, it may also participate in the process itself, by running a simulation, as a calculator, as a help facility or by playing a game.

Although there are many positive advantages to **CAL** based upon **MMI(Man-Machine-Interaction)** methods, there are obviously some disadvantages.

One major drawback is the problem of developing courseware; this is often difficult and time consuming since there are few useful courseware development tools available.

Another disadvantage of the MMI approach is the often difficult nature of the device interfacing that needs to be undertaken; when the peripheral cluster associated with a workstation contains a large number of devices, connecting and controlling them can be difficult, particularly they utilize 'non-standard' interfaces.

Yet another disadvantage of MMI is its cost; when sophisticated resources are involved such as a video disc or a complex simulation rig the cost can be prohibitively high.

HOW TEXT IS ORGANIZED

Text in this context means a collection of strings of characters or symbol taken from some alphabet and strung together according to a set of reasonably well-defined morphemic, lexical, syntactic, semantic and pragmatic rules.

The types of text may be:

♠ mathematical proofs or computations involving strings of numbers, or

♠ collections of symbols representing logical arguments, or

♠ a sequence of statements that make up a computer program, or

♠ a symbol combination that represent poetry, prose, scientific reports, or

♠ documented records of legal and juridical material.

As seen in figure 4-6 on page 141, main source of text is paper-based documents. They are stored permanently either on optical disc, or microfilm (microfiche), or in data bank (hard disk, magnetic tapes, etc.). They can be retrieved by means of software specially designed for that system. The remote learners can have access to it through electronic mailing facility, thus at the same time users or students near and far can talk(on-screen) to each other. In other words, they can do tele-conferencing.

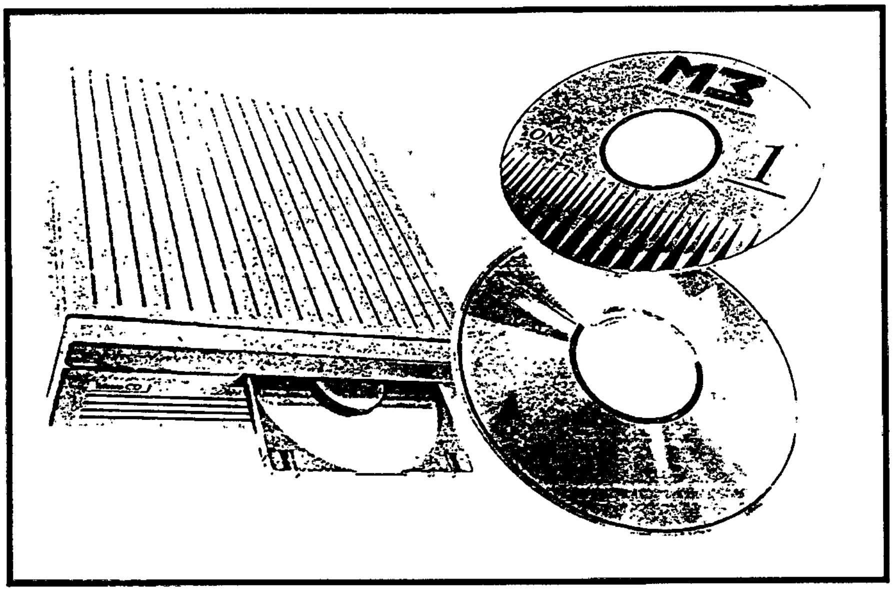
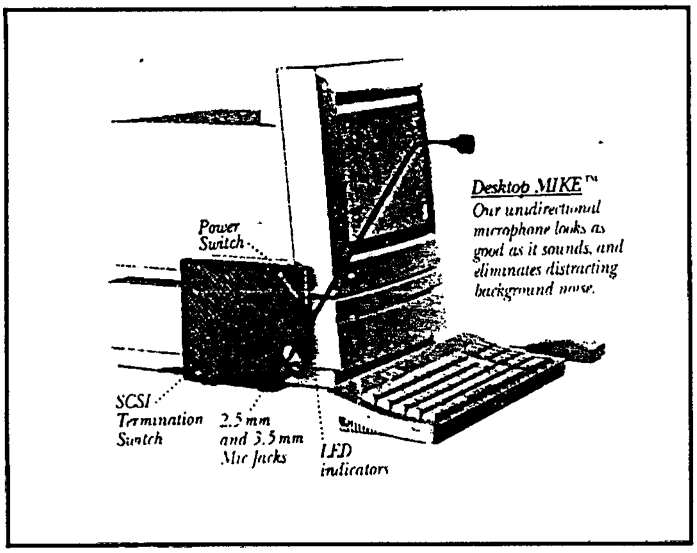


Figure 4-2. CD-ROM drive and CDs.



Desktop MIKE™
Our unidirectional microphone looks as good as it sounds, and eliminates distracting background noise.

Figure 4-3 . Voice Recognizer fitted externally to computer.

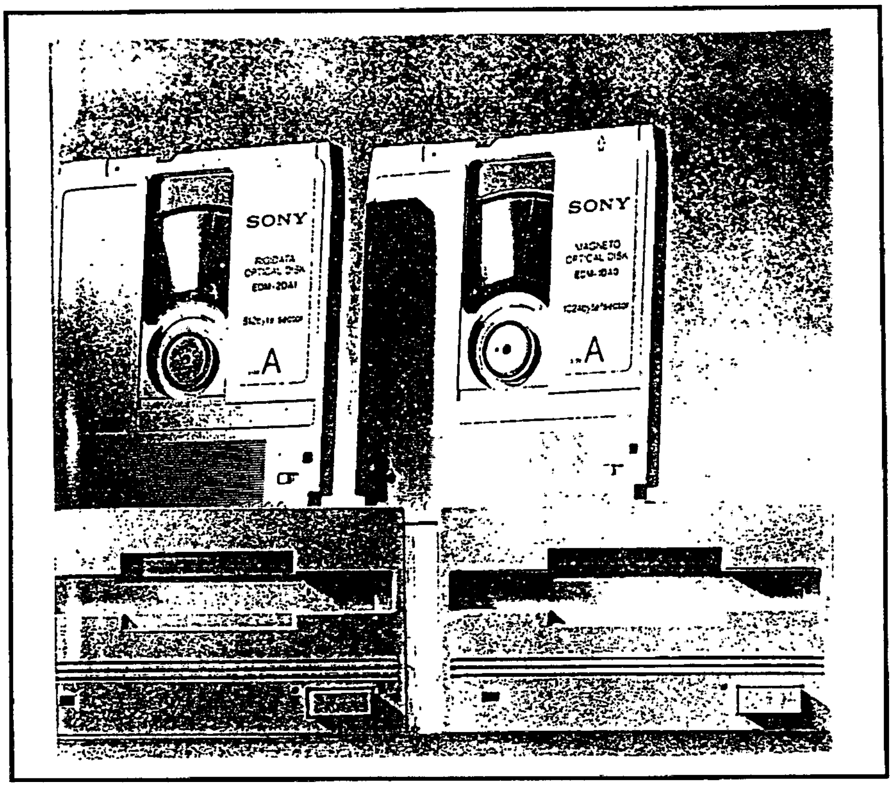


Figure 4-4. Optical Disc Drives and the Disks.

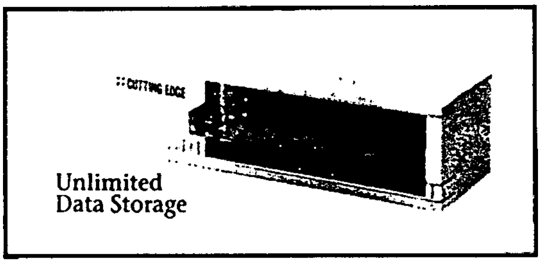


Figure 4-5. Tape backup Drive for mass data storage.

The above figures are the devices used in the Multi-media pack.

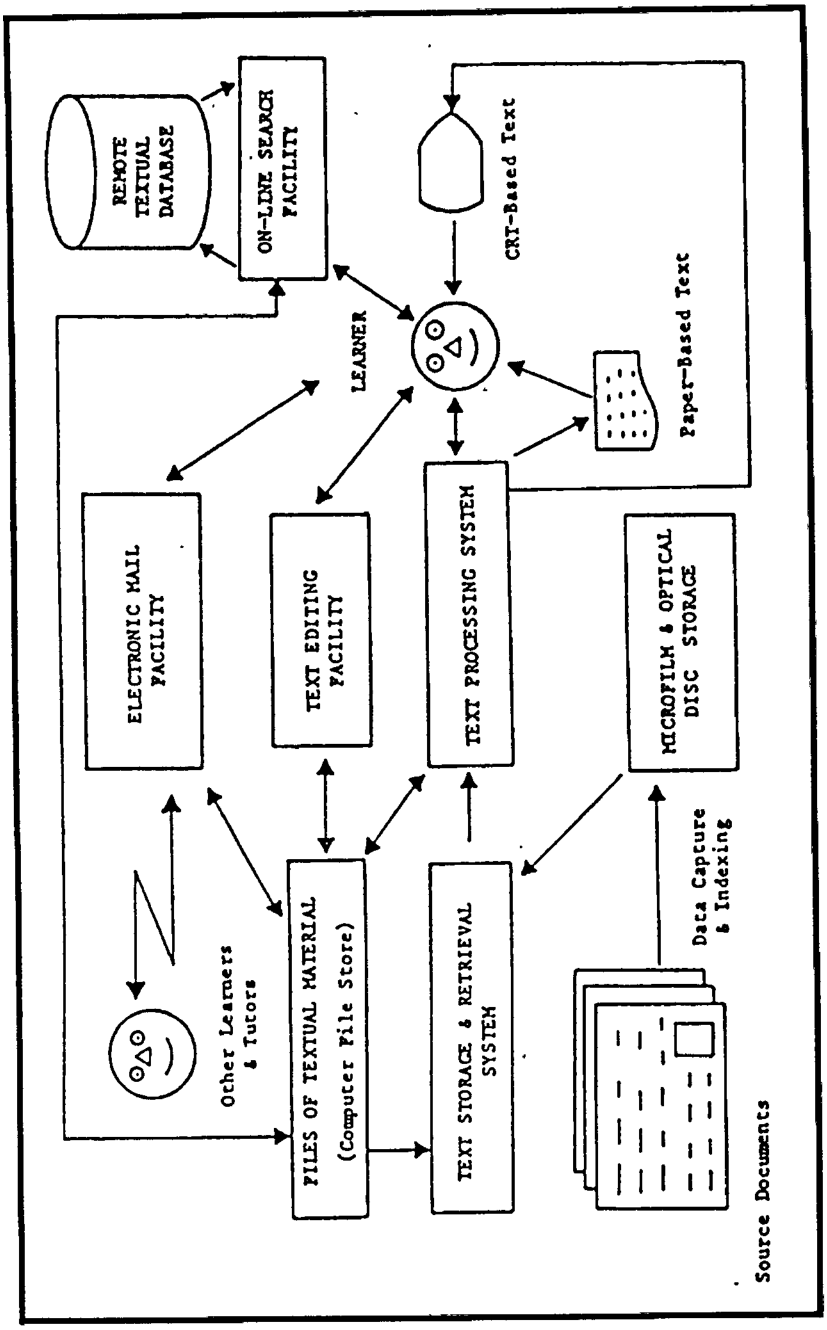


Figure 4-6. Text flow in a CAL environment.

GRAPHIC SUPPORT FOR *CAL*

Figure 4-7 illustrates schematically the way in which pictorial/graphic support might be used to augment a basic CAL dialogue involving a conventional CRT and keyboard.

There are three important ancillary (conceptual) units: an image production system, an image store, and a picture display system. Obviously, this diagram caters for a wide variety possibility. For example, the image production system could be based upon conventional manual methods of picture generation involving the use of a drawing board, paints, and/or ink. Alternatively, it might be a highly automated system involving the use of sophisticated computer graphics (perhaps in conjunction with a mouse or light pen device). The unit could equally well employ some form of camera that incorporates conventional film or video.

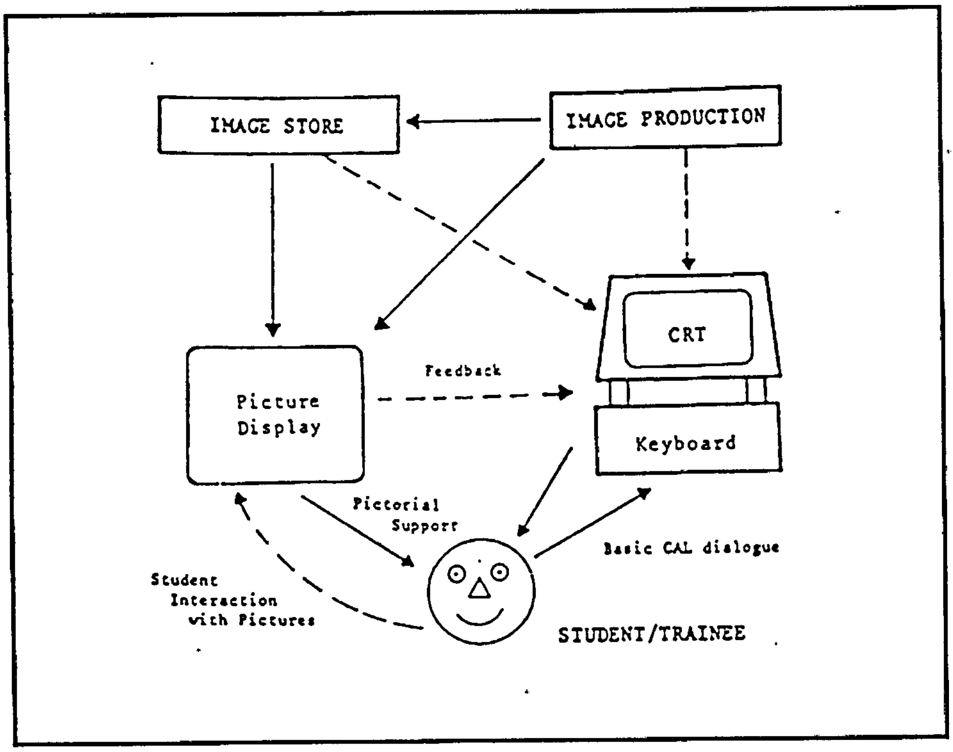


Figure 4-7.Graphic support.

When using images to support a **CAL dialogue** the instructor must decide upon the mechanism by which graphic aid will be provided. Like sound and text, graphics provides another *'logical channel'* of communication. Some thought must therefore be given to how this logical channel is to be mapped onto one or more physical support channels.

For example, should pictorial material be contained in a workbook that the student uses in conjunction with the CRT? Should a second CRT display system be used? Should the graphic support be embedded within the primary CRT/keyboard device by means of frame swapping, screen segmentation, and/or windowing techniques? Should graphic aiding be built into some other ancillary device such as a *graph plotter, a robot arm,* or a *mechanical turtle* having a 'paint brush' attached to its tail? Obviously, a large number of possibilities exist for the realization of the 'picture display system' depicted in Figure 4-7. Because of the diverse range of resources available each learning/training situation must therefore be considered in the context of its own particular pictorial aid requirements and appropriate graphic support facilities designed.

Some mention has already been made of the wide range of graphics resources available for picture production. Before concluding this section brief descriptions of some of these will be presented. Because it is not possible to offer any form of in-depth treatment of specific areas appropriate references are given (where they are needed) to more detailed descriptions of particular topics and techniques.

Because graphics has become so popular an extremely wide range of software packages now exist to support the use of this technique.

Many of these are described in the standard graphics text books cited above. These packages can be used to reduce much of the programming effort needed to generate pictures. In addition to graphics packages, many authoring languages **(Barker and Skipper, 1986)** also provide in-built facilities for picture generation.

SOUND EFFECTS

Sound effect is usually done through hardware and software. For the IBM computers, there are on the market, packages like **Sound Blaster** and **Voice Recognizer**. Software for handling the input and output of the sound and voices is also included in the package. The sound and voices can be recorded, digitized and saved as audio files via a microphone or audio input line and the interface card. The audio files that are saved in the disk can be run (playback) in many software, like PC-Storyboard.

Another parallel sound device is CD-ROM(Compact Disk-Read Only Memory). The disadvantage of the CD-ROM is its slow rate of retrieving data and the fact that it is impossibility to store new data on the disk once it is written. The advantage is that the quality of the graphics retrieved from the disk is in high resolution and sound is also incorporated in the information.

Sound medium is as powerful as the graphic medium and both demand quite a lot of RAM space and hard disk space. However, there is no doubt about the enhancement of the student learning.

CHAPTER V

CONCLUSIONS and RECOMMENDATIONS

□his dissertation has explored the potential and scope of computers in maritime education, the different means of realisation of this potential, the limitations and advantages, the methodologies of the design and development of **CBI**, as well as the ways in which microcomputers are being used in maritime education and training. The following conclusions and recommendations are to be considered against this background of work.

CONCLUSIONS

1. As the role of computers in the operational management of ships increases there will need to be a concurrent enhancement of the role of computers in maritime education, already apparent in the fields of maritime simulation.

2. There are drawbacks and limitations in the implementation of CBI. However, as a supporting element in the didactic structure of learning there is significant potential.

3. The presentation of computer based instruction is important, hence the importance of screen resolution and quality. The escalating reduction in prices of computer hardware and the increasing sophistication of software create opportunities that need to be realised by maritime educators.

4. The role of computers in the management of maritime education cannot be ignored. Testing, record keeping and time tabling are relevant areas of application.

5. Teaching programmes may use *tutorials, drills* or *simulations* for implementing **CBI**, yet the process of development must be systematic and well considered. The involvement of experienced teachers is an essential step in this process.

6. The **CBI** implementation on which this dissertation is based has demonstrated the value of using software packages in association with self developed programs.

7. Evaluation of the **CBI** implementation must commence with consideration of feedback from students who constitute the user community.

8. Currently the pool of software for maritime education and its availability is limited in scope and quality. However, there is scope for including such software in the teaching process.

9. Finally, **CBI** has a complementary potential to normal lecturing in enhancing student learning.

RECOMMENDATIONS

1. Computer literacy is an essential component of maritime education today. A needs analysis is an essential first step in its implementation.

2. Future needs should be considered before acquiring hardware. The advice of professionals in the field should be sought.

3. Software maintenance and availability are important conditions for **CBI**. Controlled but ready access through networking is essential.

4. Computer hardware should not become a hierarchical token. Maximising availability to those involved in development or application is essential.

5. Training of staff in the use of computers is essential, so that productivity can be maximised, both in the office and the classroom.

6. The provision of relevant literature for staff development is important. Conference publications such as those of **IMLA**, **INSLC** should be made available to staff.

7. The involvement of specialist programming staff in maritime institutions is to be recommended so that teachers and programmers can interact in a synergistic manner to maximise the quality of the output and minimise the time involved.

8. There is much scope for regional and international co-operation in this area. Organisations and associations in the fields of computers and informatics can

be valuable partners in advancing the implementation of **CBI** in maritime institutions.

Reference List

ADAMS M., & BRUCE, B. 1980. "*Background Knowledge and Reading Comprehension*", Reading Education Report No.13. Urbana: Center for the Study of Reading. University of Illinois at Urbana-Champaign.

ALESSI, S.M. & STANLEY R. TROLLIP. 1985. Computer Based Instruction Methods And Development. USA.

ANDERSON, J.R. 1976. Language, Memory. and Thought. HilIsdale, NJ: Lawrence Erlbaum Associates.

ANDERSON, R.C. 1977. The Notion of Schemata and the Educational Enterprise, in Schooling and Acquisition of Knowledge. eds. R.C.

ANDERSON, R.J. SPIRO, & W.E. MONTAGUE. Hillsdale, NJ: Lawrence Erlbaum Associates.

ANDERSON, R.C. & BIDDLE, W.B. 1975. "*On Asking People Questions About What They Are Reading*", in Psychology of Learning and Motivation (vol. 9), ed. G. BOWER. NEWYORK: Academic Press.

BARKER, P. 1989. Multi-media Computer Assisted Learning. London: Kogan Page Ltd.

BARKER, P.G. and SKIPPER, T. 1986. "*A Practical Introduction To Authoring For Computer Assisted Instruction", Part 7: Graphic Support For CAL,* British Journal Of Educational Technology, 17(3),, 194-212.

BEARD, R.M. & J. HARTLY. 1984. Teaching And Learning In Higher Education. 4th ed. London: Paul Chapman Publishing Ltd.

DAVID GLASS. 1992. "*Looking For Security Ashore*", Seatrade Review (February, 1992). Colchester: Seatrade Publication Ltd.

DEAN, C, & W. QUENTINE. 1989. A Hand Book Of Computer Based Training. 2nd. Ed. New York: Nichols Publishing.

DURRET, J., & TREZONA, J. 1982 "*How To Use Color Displays Effectively: The Elements of Color Vision and Their Implications for Programmers,*" Pipeline, 7, no. 2 , 13-16.

DWYER, F.M. 1978. Strategies for lmproving Visual Learning. State College, PA: Learning Services.

ELLINGTON, H. 1990. Teaching Through Self-Instruction. London: Kogan Page. first published Kogan Page, 1986.

FLEMING, M., & LEVIE, W.H. 1978. Instructional Message Design: Principles from the Behavioural Sciences. Englewood Cliffs, NJ: Educational Technology Publications.

GAGNE, R.M., WAGER, W., & ROJAS, A. *"Planning and Authoring Computer Assisted Instruction Lessons,"* Educational Technology (September, 1981),17-31.

GARLAND, J, & MUNN, C. 1983. *Micro-computer And Teaching Biology,* CAL News 21, February.

GAYNOR, P. 1981. "*The Effect of Feedback Delay on Retention of Computer-based Mathematical Material",* Journal of Computer-based Instruction, 8, 28-34.

HEINES, JESSE M. 1984 Screen Design Strategies For Computer-assisted Learning. Bedford, Mass: Digital Press.

HOFMANN, L. 1991. "*Computers in Education: A Triumph of Process Over Purpose?"* Collegiate Microcomputer, vol ix, No. 4. Rose-Hulman Institute of Technology.

JAY, T. 1981. "*Computerphobia: What To Do A Bout It",* Educational Technology,21: 47-48.

KLAUSMEIER, H.J. & FELDMAN, K.V. 1975. "*Effect of a Definition and a Varying Number of Examples and Nonexamples On Concept Attainment",* Journal of Educational Psychology, 67, 174-78.

KORAN, M.L. 1971. "*Differential Response to Inductive and Deductive Instructional Procedures",* Journal of Educational Psychology, 62, 300-307.

LARRY, D, ROSEN & M. M. WEIL. 1990. "*Computers, Classroom Instruction, And The Computerphobic University Student",* Collegiate Microcomputer, vol. VIII, No. 4.

LYE, J.D. 1991. *"Computer Assisted Identification keys In Place of Dichotomous Keys for Microbiology Laboritories,"* Collegiate Microcomputers, vol. iv, No.3(August, 91). Indiana: Rose-Hulman Institute of Technology.

MARKLE, S.M. 1969. Good Frames and Bad: A Grammar of Frame Writing. New York: John Willy.

McCABE, D. 1986. *"Simulation and Games in Teaching & Learning"*, Teaching Methodology Papers. Hong Kong: Polytechnic Education Technology Unit.

****Megarry Jacquetta et al.** 1983. Computers and Education. London: Kogan Page, p.50.

******** **Ibid.**, p.55.

MERRILL, P.F. 1974. *"Effects of the Availability of Objectives and/or Rules on the Learning Process"*, Journal of Educational Psychology, 66, 534-39.

ODEY, S and R. Facey. 1988. *"The IMO STCW 1978 - Standardising the Standard"*, IMLA Newsletter 13 (March, 1988).

RIGNEY, J.W. & LUTZ, K.A. 1976. *"Effect of Graphic Analogies of Concepts in Chemistry on Learning and Attitude"*, Journal of Education Psychology. 68.

ROWNTREE, DEREK. 1990. Teaching Through Self Instruction. Revised edition. First published in 1986. New York: Nichols Publishing.

SINGH, J.S. 1988. *"Microprocessors and Simulators: What Contribution Can They Make To The Efficiency and Effectiveness of The Training of Shipboard Personnel?"* Conference Paper of Fifth International Maritime Education and Training. The International Maritime Lecturers Association.

***TANSEY, P.J.** 1971. Educational Aspects of Simulation. London: McGraw-Hill, p.226.

ATTACHMENT (1)

PROGRAM LISTING FOR "REGISTER.BAS" FILE

```
10 CLS : SCREEN 0: KEY OFF: COLOR 9, 9: CLS : COLOR 6, 7
20 LOCATE 2, 27: PRINT STRING$(1, 32); : PRINT STRING$(1, 201);
30 PRINT STRING$(25, 205); : PRINT STRING$(1, 187); : PRINT STRING$(1, 32)
40 LOCATE 3, 27: PRINT STRING$(1, 32); : PRINT STRING$(1, 186);
50 PRINT STRING$(25, 32); : PRINT STRING$(1, 186); : PRINT STRING$(1, 32)
60 LOCATE 4, 27: PRINT STRING$(1, 32); : PRINT STRING$(1, 200);
70 PRINT STRING$(25, 205); : PRINT STRING$(1, 188); : PRINT STRING$(1, 32)
80 COLOR 0, 0
90 LOCATE 3, 56: PRINT STRING$(2, 177)
100 LOCATE 4, 56: PRINT STRING$(2, 177)
110 LOCATE 5, 29: PRINT STRING$(29, 177)
120 COLOR 6, 7
130 LOCATE 7, 6: PRINT STRING$(1, 32); : PRINT STRING$(1, 201);
140 PRINT STRING$(64, 205); : PRINT STRING$(1, 187); : PRINT STRING$(1, 32)
150 FOR I = 8 TO 21
160 LOCATE I, 6: PRINT STRING$(1, 32); : PRINT STRING$(1, 186);
170 PRINT STRING$(64, 32); : PRINT STRING$(1, 186); : PRINT STRING$(1, 32)
180 NEXT I
190 LOCATE 22, 6: PRINT STRING$(1, 32); : PRINT STRING$(1, 200);
200 PRINT STRING$(40, 205);
210 PRINT STRING$(24, 205); : PRINT STRING$(1, 188); : PRINT STRING$(1, 32)
220 FOR I = 8 TO 23
230 COLOR 0, 0: LOCATE I, 74: PRINT STRING$(2, 177)
240 NEXT I
250 LOCATE 23, 8: PRINT STRING$(68, 177)
260 LOCATE 3, 32: COLOR 5,7: PRINT "LIFEBOAT EQUIPMENTS"
270 LOCATE 7, 26: COLOR 14, 6: PRINT "  STUDENT REGISTRATION INPUT  "
280 LOCATE 7, 56: COLOR 0, 7: PRINT STRING$(1, 220)
290 LOCATE 8, 27: PRINT STRING$(30, 223)
300 DAT$ = DATE$: LOCATE 10, 12: COLOR 15, 2: PRINT " Today Date : "; DATE$; " "
310 LOCATE 10, 37: COLOR 0, 7: PRINT "Ü"
320 LOCATE 11, 13: COLOR 0, 7: PRINT STRING$(25, 223)
330 TIM$ = TIME$: LOCATE 10, 40: COLOR 15, 2: PRINT " Registered Time : "; TIME$;
        " "
340 LOCATE 10, 68: COLOR 0, 7: PRINT "Ü"
350 LOCATE 11, 41: COLOR 0, 7: PRINT STRING$(28, 223)
360 LOCATE 13, 17: COLOR 15,6: PRINT "    Name : "
370 LOCATE 13, 28: COLOR 0, 7: PRINT "Ü"
380 LOCATE 14, 18: COLOR 0, 7: PRINT STRING$(11, 223)
390 LOCATE 13, 32: COLOR 14, 3: PRINT "                                    "
400 LOCATE 13, 68: COLOR 0, 7: PRINT "Ü"
410 LOCATE 14, 33: COLOR 0, 7: PRINT STRING$(36, 223)
420 LOCATE 15, 17: COLOR 15, 6: PRINT "  Id no. : "
430 LOCATE 15, 28: COLOR 0, 7: PRINT "Ü"
440 LOCATE 16, 18: COLOR 0, 7: PRINT STRING$(11, 223)
450 LOCATE 15, 32: COLOR 14, 3: PRINT "                                    "
460 LOCATE 15, 68: COLOR 0, 7: PRINT "Ü"
```

```
470 LOCATE 16, 33: COLOR 0, 7: PRINT STRING$(36, 223)
480 LOCATE 17, 17: COLOR 15, 6: PRINT "    Year : "
490 LOCATE 17, 28: COLOR 0, 7: PRINT "Ü"
500 LOCATE 18, 18: COLOR 0, 7: PRINT STRING$(11, 223)
510 LOCATE 17, 32: COLOR 14, 3: PRINT "                                    "
520 LOCATE 17, 68: COLOR 0, 7: PRINT "Ü"
530 LOCATE 18, 33: COLOR 0, 7: PRINT STRING$(36, 223)
540 LOCATE 19, 17: COLOR 15, 6: PRINT "   Class : "
550 LOCATE 19, 28: COLOR 0, 7: PRINT "Ü"
560 LOCATE 20, 18: COLOR 0, 7: PRINT STRING$(11, 223)
570 LOCATE 19, 32: COLOR 14, 3: PRINT "                                    "
580 LOCATE 19, 68: COLOR 0, 7: PRINT "Ü"
590 LOCATE 20, 33: COLOR 0, 7: PRINT STRING$(36, 223)
600 LOCATE 13, 32: COLOR 15, 3: INPUT "", NAM$
610 LE = LEN(NAM$)
620 IF LE > 29 THEN GOSUB 860: GOTO 350
630 LOCATE 15, 32: COLOR 15, 3: INPUT "", IDN$
640 IDS%= VAL(IDN$)
650 IF IDS%< 1 OR IDS%> 10000 THEN GOSUB 900: GOTO 630
660 LOCATE 17, 32: COLOR 15, 3: INPUT "", YEA$
670 LOCATE 19, 32: COLOR 15, 3: INPUT "", CLA$
680 COLOR 0, 1
690 OPEN "R", #1, "SCORE.DAT", 80
700 FIELD #1, 29 AS NA$, 5 AS ID$, 10 AS YE$, 15 AS CL$, 3 AS SC$, 10 AS DA$, 8
        AS TI$
710 LSET NA$ = NAM$
720 LSET ID$ = IDN$
730 LSET YE$ = YEA$
740 LSET CL$ = CLA$
750 LSET DA$ = DAT$
760 LSET TI$ = TIM$
770 PUT #1, IDS%
780 CLOSE #1
790 CLS
800 OPEN "R", #2, "temp.dat", 5
810 FIELD #2, 5 AS ID$
820 LSET ID$ = IDN$
830 PUT #2, 1
840 CLOSE #2
850 END
860 LOCATE 21, 15: COLOR 14, 7: PRINT " Name characters entered must not be more
        than 29. "
870 FOR I = 1 TO 10000: NEXT I
880 LOCATE 21, 15: COLOR 7, 7: PRINT STRING$(55, 32)
890 RETURN
900 LOCATE 21, 15: COLOR 14, 7: PRINT " Id no. entered must be a number between 1
        to 10000 "
910 FOR I = 1 TO 10000: NEXT I
920 LOCATE 21, 15: COLOR 7, 7: PRINT STRING$(55, 32)
930 LOCATE 15, 32: COLOR 14, 3: PRINT "                   "
940 RETURN
950 END
```

ATTACHMENT (2)

PROGRAM LISTING FOR "TEST.BAS" FILE

```
10 KEY OFF: SCREEN 0: COLOR 9, 1: CLS : COLOR 0, 7
20 TIMER ON: MIN = 10: SEC = 0
30 ON TIMER(1) GOSUB 5001
40  LOCATE  2,  13:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      201);
50  PRINT  STRING$(53,  205);  :  PRINT  STRING$(1,  187);  :  PRINT
      STRING$(1, 32)
60  LOCATE  3,  13:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      186);
70  PRINT  STRING$(53,  32);  :  PRINT  STRING$(1,  186);  :  PRINT
      STRING$(1, 32)
80  LOCATE  4,  13:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      200);
90  PRINT  STRING$(53,  205);  :  PRINT  STRING$(1,  188);  :  PRINT
      STRING$(1, 32)
100 COLOR 0, 0
110 LOCATE 3, 70: PRINT STRING$(2, 177)
120 LOCATE 4, 70: PRINT STRING$(2, 177)
130 LOCATE 5, 15: PRINT STRING$(57, 177)
140 COLOR 0, 7
150  LOCATE  7,  6:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      201);
160  PRINT  STRING$(64,  205);  :  PRINT  STRING$(1,  187);  :  PRINT
      STRING$(1, 32)
170 FOR I = 8 TO 17
180  LOCATE  I,  6:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      186);
190  PRINT  STRING$(64,  32);  :  PRINT  STRING$(1,  186);  :  PRINT
      STRING$(1, 32)
200 NEXT I
210  LOCATE  18,  6:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      199);
220 PRINT STRING$(40, 196); : PRINT STRING$(1, 194);
230  PRINT  STRING$(23,  196);  :  PRINT  STRING$(1,  182);  :  PRINT
      STRING$(1, 32)
240 FOR I = 19 TO 21
250  LOCATE  I,  6:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      186);
260  PRINT  STRING$(40,  32);  :  PRINT  STRING$(1,  179);  :  PRINT
      STRING$(23, 32);
270 PRINT STRING$(1, 186); : PRINT STRING$(1, 32)
280 NEXT I
290  LOCATE  22,  6:  PRINT  STRING$(1,  32);  :  PRINT  STRING$(1,
      200);
300 PRINT STRING$(40, 205); : PRINT STRING$(1, 207);
310  PRINT  STRING$(23,  205);  :  PRINT  STRING$(1,  188);  :  PRINT
      STRING$(1, 32)
```

```
320 FOR I = 8 TO 23
330 COLOR 0, 0: LOCATE I, 74: PRINT STRING$(2, 177)
340 NEXT I
350 LOCATE 23, 8: PRINT STRING$(68, 177)
360 COLOR 1, 15: LOCATE 3, 34: PRINT " TEST PROGRAM "
370 COLOR 0, 10: LOCATE 7, 34: PRINT "  QUESTION  "
380 COLOR 0, 7: LOCATE 8, 35: PRINT STRING$(12, 223)
390 LOCATE 7, 46: PRINT STRING$(1, 220)
400 COLOR 0, 10: LOCATE 18, 22: PRINT "  MESSAGE  "
410 COLOR 0, 7: LOCATE 19, 23: PRINT STRING$(11, 223)
420 LOCATE 18, 33: PRINT STRING$(1, 220)
430 COLOR 0, 10: LOCATE 18, 56: PRINT "  SCORE  "
440 COLOR 0, 7: LOCATE 19, 57: PRINT STRING$(9, 223)
450 LOCATE 18, 65: PRINT STRING$(1, 220)
460 COLOR 1, 15
470 SO = 0: NO = 0: DIM RIGHT(10), WRONG(10): TI2$ = TIME$
480 RANDOMIZE TIMER
490 T = INT(RND * 10 + 1)
500 IF T = 1 THEN RESTORE 3690
510 IF T = 2 THEN RESTORE 3700
520 IF T = 3 THEN RESTORE 3710
530 IF T = 4 THEN RESTORE 3720
540 IF T = 5 THEN RESTORE 3730
550 IF T = 6 THEN RESTORE 3740
560 IF T = 7 THEN RESTORE 3750
570 IF T = 8 THEN RESTORE 3760
580 IF T = 9 THEN RESTORE 3770
590 IF T = 10 THEN RESTORE 3780
600 GOSUB 3590
610 FOR LO = 1 TO 10
620 READ QU
630 GOSUB 3490
640 LOCATE 9, 8: PRINT LO; : COLOR 4, 7: PRINT ">"; : COLOR 1,
      7
650 ON QU GOSUB 1560, 1680, 1800, 1950, 2080, 2210, 2350,
      2500, 2660, 2780
660 NEXT LO
670 OPEN "R", #2, "temp.dat", 5
680 FIELD #2, 5 AS ID$
690 LSET ID$ = IDN$
700 GET #2, 1
710 ID = VAL(ID$)
720 CLOSE #2
730 OPEN "R", #1, "SCORE.DAT", 80
740 FIELD #1, 29 AS NA$, 5 AS ID$, 10 AS YE$, 15 AS CL$, 3 AS
      SC$, 10 AS DA$, 8 AS TI$
750 GET #1, ID
760 CLS
770 COLOR 7, 7: CLS
780 FOR I = 1 TO 25
790 PRINT STRING$(80, 32);
800 NEXT I
810 LOCATE 1, 1: COLOR 10, 2: PRINT STRING$(80, 32)
820 LOCATE 1, 33: COLOR 0, 2: PRINT " SCORE REPORT "
```

```
830 LOCATE 3, 4: COLOR 1, 7: PRINT "Name : ";
840 LOCATE 3, 11: COLOR 5, 7: PRINT NA$
850 LOCATE 3, 43: COLOR 1, 7: PRINT "Id no. : ";
860 LOCATE 3, 51: COLOR 5, 7: PRINT ID$
870 LOCATE 4, 4: COLOR 1, 7: PRINT "Year : ";
880 LOCATE 4, 11: COLOR 5, 7: PRINT YE$
890 LOCATE 4, 43: COLOR 1, 7: PRINT "Class  : ";
900 LOCATE 4, 51: COLOR 5, 7: PRINT CL$
910 LOCATE 5, 4: COLOR 1, 7: PRINT "Date of studying : ";
920 LOCATE 5, 23: COLOR 5, 7: PRINT DA$
930 LOCATE 5, 43: COLOR 1, 7: PRINT "Starting time : ";
940 LOCATE 5, 59: COLOR 5, 7: PRINT TI$
950 LOCATE 6, 4: COLOR 1, 7: PRINT "Date of testing  : ";
960 LOCATE 6, 23: COLOR 5, 7: PRINT DATE$
970 LOCATE 6, 43: COLOR 1, 7: PRINT "Starting time : ";
980 LOCATE 6, 59: COLOR 5, 7: PRINT TI2$
990 LOCATE 7, 4: COLOR 1, 7: PRINT "Time spent on lessons : ";
1000 TI1$ = TI$
1010 H1 = VAL(LEFT$(TI1$, 2))
1020 M1 = VAL(MID$(TI1$, 4, 2))
1030 S1 = VAL(RIGHT$(TI1$, 2))
1040 H2 = VAL(LEFT$(TI2$, 2))
1050 : M2 = VAL(MID$(T12$, 4, 2))
1060 S2 = VAL(RIGHT$(TI2$, 2))
1070 IF H2 < H1 THEN H = (24 - H1) + H2 ELSE H = H2 - H1
1080 IF M2 < M1 THEN M = (60 - M1) + M2: H = H - 1 ELSE M = M2
     - M1
1090 IF S2 < S1 THEN S = (60 - S1) + S2: M = M - 1 ELSE S = S2
     - S1
1100 LOCATE 7, 31: COLOR 5, 7: PRINT H;
1110 COLOR 1, 7: PRINT "hour(s)";
1120 COLOR 5, 7: PRINT M;
1130 COLOR 1, 7: PRINT "minute(s)";
1140 COLOR 5, 7: PRINT S;
1150 COLOR 1, 7: PRINT "second(s)"
1160 T13$ = TIME$
1170 H3 = VAL(LEFT$(TI3$, 2))
1180 M3 = VAL(MID$(TI3$, 4, 2))
1190 S3 = VAL(RIGHT$(TI3$, 2))
1200 IF H3 < H2 THEN H = (24 - H2) + H3 ELSE H = H3 - H2
1210 IF M3 < M2 THEN M = (60 - M2) + M3: H = H - 1 ELSE M = M3
     - M2
1220 IF S3 < S2 THEN S = (60 - S2) + S3: M = M - 1 ELSE S = S3
     - S2
1230 LOCATE 8, 4: COLOR 1, 7: PRINT "Time spent on test     :
     ";
1240 LOCATE 8, 31: COLOR 5, 7: PRINT H;
1250 COLOR 1, 7: PRINT "hour(s)";
1260 COLOR 5, 7: PRINT M;
1270 COLOR 1, 7: PRINT "minute(s)";
1280 COLOR 5, 7: PRINT S;
1290 COLOR 1, 7: PRINT "second(s)"
1300 LOCATE 8, 7: COLOR 2, 7: PRINT STRING$(80, 205)
1310 LOCATE 10, 10: COLOR 0, 7: PRINT "Item"
```

```
1320 LOCATE 10, 30: COLOR 0, 7: PRINT "Right"
1330 LOCATE 10, 50: COLOR 0, 7: PRINT "Wrong"
1340 LOCATE 11, 1: COLOR 2, 7: PRINT STRING$(80, 205)
1350 FOR I = 1 TO 10
1360  LOCATE  11  +  I,  10:  COLOR  0,  7:  PRINT  I;  TAB(30);
       RIGHT(I); TAB(50); WRONG(I)
1370 NEXT I
1380 LOCATE 22, 1: COLOR 2, 7: PRINT STRING$(80, 205)
1390 COLOR 1: LOCATE 23, 25: PRINT "Sum";
1400 LOCATE 23, 30: COLOR 5, 7: PRINT R;
1410 LOCATE 23, 45: COLOR 1, 7: PRINT "Sum";
1420 LOCATE 23, 50: COLOR 5, 7: PRINT W
1430 COLOR 1: LOCATE 16, 62: PRINT "Average "; : COLOR 5, 7:
       PRINT R * 10;
1440 R = R * 10
1450 R$ = STR$(R)
1460 LSET SC$ = R$
1470 PUT #1, ID
1480 COLOR 1, 7: PRINT " %"
1490 LOCATE 25, 1: COLOR 2, 2: PRINT STRING$(80, 32);
1495 LOCATE 25, 27: COLOR 4, 2: PRINT " To printer (y/n) ?";
1500 A$ = INKEY$: TIMER OFF
1510 IF A$ = "y" OR A$ = "Y" THEN 5020
1520 IF A$ = "n" OR A$ = "N" THEN END
1525 IF A$ <> " " THEN 1500
1530 COLOR 7, 0: CLS
1540 CLOSE #1
1550 END
1560 PRINT " What equipment is used for holding the life boat
       in the "
1565 LOCATE 10, 10: PRINT "   water? "
1570 LOCATE 12, 15: PRINT "a) Heliograph "
1580 LOCATE 13, 15: PRINT "b) Painter "
1590 LOCATE 14, 15: PRINT "c) Sea anchor "
1600 LOCATE 15, 15: PRINT "d) Oar "
1610 SOLU$ = "c"
1620 WRO1$ = "b"
1630 WRO2$ = "a"
1640 WRO3$ = "d"
1650 GOSUB 3630
1660 GOSUB 2920
1670 RETURN
1680 PRINT " What equipment is used to facilitate towing? "
1690 LOCATE 11, 15: PRINT "a) Heaving line "
1700 LOCATE 12, 15: PRINT "b) Lifeline "
1710 LOCATE 13, 15: PRINT "c) Fishing kit "
1720 LOCATE 14, 15: PRINT "d) Painter "
1730 SOLU$ = "d"
1740 WRO1$ = "a"
1750 WRO2$ = "b"
1760 WRO3$ = "c"
1770 GOSUB 3630
1780 GOSUB 2920
1790 RETURN
```

```
1800 PRINT " When will be the pyrotechnics expired? "
1810 LOCATE 11, 15: PRINT "a) 2 years after date of
      manufacture"
1820 LOCATE 12, 15: PRINT "b) 3 years after date of
      manufacture"
1830 LOCATE 13, 15: PRINT "c) 4 years after date of
      manufacture "
1840 LOCATE 14, 15: PRINT "d) 5 years after date of
      manufacture"
1880 SOLU$ = "b"
1890 WRO1$ = "a"
1900 WRO2$ = "c"
1910 WRO3$ = "d"
1920 GOSUB 3630
1930 GOSUB 2920
1940 RETURN
1950 PRINT " How many thermal protective aids shall be carried
      in a "
1960 LOCATE 10, 10: PRINT "   lifeboat?"
1970 LOCATE 12, 15: PRINT "a) 10 % of compliment of lifeboat"
1980 LOCATE 13, 15: PRINT "b) 2 units  "
1990 LOCATE 14, 15: PRINT "c) 3 units       "
2000 LOCATE 15, 15: PRINT "d) at least 3"
2010 SOLU$ = "a"
2020 WRO1$ = "b"
2030 WRO2$ = "c"
2040 WRO3$ = "d"
2050 GOSUB 3630
2060 GOSUB 2920
2070 RETURN
2080 PRINT " The use of fish or vegetable oil is:"
2090 LOCATE 11, 15: PRINT "a) to reserve as emergency food"
2100 LOCATE 12, 15: PRINT "b) to burn in the oil lantern"
2110 LOCATE 13, 15: PRINT "c) to reduce the breaking of waves"
2120 LOCATE 14, 15: PRINT "d) to apply on the body to prevent
      sunburns"
2140 SOLU$ = "c"
2150 WRO1$ = "a"
2160 WRO2$ = "b"
2170 WRO3$ = "d"
2180 GOSUB 3630
2190 GOSUB 2920
2200 RETURN
2210 PRINT " What is the equipment used for distress signal?"
2220 LOCATE 11, 15: PRINT "a) Torch"
2230 LOCATE 12, 15: PRINT "b) Lantern"
2240 LOCATE 13, 15: PRINT "c) Rocket parachute"
2250 LOCATE 14, 15: PRINT "d) Heliograph"
2280 SOLU$ = "c"
2290 WRO1$ = "a"
2300 WRO2$ = "b"
2310 WRO3$ = "d"
2320 GOSUB 3630
2330 GOSUB 2920
```

```
2340 RETURN
2350 PRINT " The use of hatchet is:"
2360 LOCATE 11, 15: PRINT "a) to protect the incharge of the
       life boat"
2370 LOCATE 12, 15: PRINT "     from being attacked by the
       crew."
2380 LOCATE 13, 15: PRINT "b) to protect the crew from those
       who lost"
2390 LOCATE 14, 15: PRINT " .  control."
2400 LOCATE 15, 15: PRINT "c) to cut the painter in case of
       emergency. "
2410 LOCATE 16, 15: PRINT "d) to cut the mast in emergency
       when the "
2420 LOCATE 17, 15: PRINT "    boat is under sail ."
2430 SOLU$ = "c"
2440 WRO1$ = "b"
2450 WRO2$ = "a"
2460 WRO3$ = "d"
2470 GOSUB 3630
2480 GOSUB 2920
2490 RETURN
2500 PRINT " Normally the size and length of the hawser is:"
2520 LOCATE 11, 15: PRINT "a)  2 inches in circum. and 2 times
       the boatlength."
2530 LOCATE 12, 15: PRINT "b)  2 and a half inch in circum.
       and 3 times the "
2540 LOCATE 13, 15: PRINT "    boatlength."
2550 LOCATE 14, 15: PRINT "c)  2 and 3 quarter inch in circum.
       and 3 times "
2560 LOCATE 15, 15: PRINT "    the boatlength."
2570 LOCATE 16, 15: PRINT "d)  3 inch in circum. and 3 times
       the boatlength. "
2590 SOLU$ = "d"
2600 WRO1$ = "a"
2610 WRO2$ = "b"
2620 WRO3$ = "c"
2630 GOSUB 3630
2640 GOSUB 2920
2650 RETURN
2660 PRINT " The compass of the life boat must be:"
2670 LOCATE 11, 15: PRINT "a) fixed so that its lubber line is
       in line with"
2680 LOCATE 12, 15: PRINT "    the fore and aft direction the
       boat."
2690 LOCATE 13, 15: PRINT "b) fixed so that its North is in
       line with "
2700 LOCATE 14, 15: PRINT "     the fore and aft line the
       boat."
2701 LOCATE 15, 15: PRINT "c) fixed close to the bow of the
       boat."
2702 LOCATE 16, 15: PRINT "d) fixed at a high place on the
       boat."
2710 SOLU$ = "a"
2720 WRO1$ = "c"
```

```
2730 WRO2$ = "b"
2740 WRO3$ = "d"
2750 GOSUB 3630
2760 GOSUB 2920
2770 RETURN
2780 PRINT " The purpose of the thermal protective aid is:"
2790 LOCATE 11, 15: PRINT "a) to reduce the fever of a sick
      person."
2800 LOCATE 12, 15: PRINT "b) to reduce the body heat loss of
      a person wearing"
2810 LOCATE 13, 15: PRINT "   it in an extremely cold water."
2820 LOCATE 14, 15: PRINT "c) to increase the body temperature
      of a person"
2830 LOCATE 15, 15: PRINT "   wearing it."
2831 LOCATE 16, 15: PRINT "d) to float on the water."
2840 SOLU$ = "b"
2850 WRO1$ = "a"
2860 WRO2$ = "c"
2870 WRO3$ = "d"
2880 GOSUB 3630
2890 GOSUB 2920
2900 RETURN
2910 END
2920 RANDOMIZE TIMER
2930 QA = INT(RND * 6 + 1)
2940 N = 0
2950 K$ = INKEY$
2960 IF K$ = SOLU$ THEN SO = SO + 1: RIGHT(LO) = 1: WRONG(LO)
      = 0: R = R + 1: GOSUB 3030
2970 IF K$ = WRO1$ THEN SO = SO + 0: RIGHT(LO) = 0: WRONG(LO)
      = 1: W = W + 1: GOSUB 3130
2980 IF K$ = WRO2$ THEN SO = SO + 0: RIGHT(LO) = 0: WRONG(LO)
      = 1: W = W + 1: GOSUB 3130
2990 IF K$ = WRO3$ THEN SO = SO + 0: RIGHT(LO) = 0: WRONG(LO)
      = 1: W = W + 1: GOSUB 3130
3000 IF N > 0 THEN RETURN
3010 IF K$ <> " " THEN 2950
3020 RETURN
3030 ON QA GOSUB 3250, 3270, 3290, 3310, 3330, 3350
3040 N = 1
3050 LOCATE 20, 15: PRINT STRING$(31, 32)
3060 LOCATE 20, 15: COLOR 5, 7: PRINT M$
3070 GOSUB 3590
3080 FOR I = 500 TO 2000 STEP 500
3090 SOUND I, 1
3100 NEXT I
3110 FOR I = 1 TO 5000: NEXT I
3120 RETURN
3130 ON QA GOSUB 3370, 3390, 3410, 3430, 3450, 3470
3140 N = 1
3150 LOCATE 20, 15: PRINT STRING$(31, 32)
3160 LOCATE 20, 15: COLOR 5, 7: PRINT M$
3170 GOSUB 3590
3180 FOR J = 1 TO 3
```

```
3190 FOR I = 2000 TO 500 STEP -500
3200 SOUND I, .8
3210 NEXT I
3220 NEXT J
3230 FOR I = 1 TO 5000: NEXT I
3240 RETURN
3250 M$ = "Congratulation! You're right."
3260 RETURN
3270 M$ = "Very good! "
3280 RETURN
3290 M$ = "Well done! "
3300 RETURN
3310 M$ = "You got it"
3320 RETURN
3330 M$ = "That is correct."
3340 RETURN
3350 M$ = "Great! "
3360 RETURN
3370 M$ = "Sorry! That's wrong"
3380 RETURN
3390 M$ = "That is not correct"
3400 RETURN
3410 M$ = "That is not right"
3420 RETURN
3430 M$ = "Try again!"
3440 RETURN
3450 M$ = "Do it again !"
3460 RETURN
3470 M$ = "You need to learn more."
3480 RETURN
3490 LOCATE 9, 8: PRINT STRING$(60, 32)
3500 LOCATE 10, 8: PRINT STRING$(60, 32)
3510 LOCATE 11, 8: PRINT STRING$(60, 32)
3520 LOCATE 12, 8: PRINT STRING$(60, 32)
3530 LOCATE 13, 8: PRINT STRING$(60, 32)
3540 LOCATE 14, 8: PRINT STRING$(60, 32)
3550 LOCATE 15, 8: PRINT STRING$(60, 32)
3560 LOCATE 16, 8: PRINT STRING$(60, 32)
3570 LOCATE 17, 8: PRINT STRING$(60, 32)
3580 RETURN
3590 LOCATE 20, 52: COLOR 1, 7: PRINT "You get"; : COLOR 11,
       7: PRINT SO;
3600 COLOR 1, 7: PRINT "from"; : COLOR 11, 7: PRINT NO: COLOR
       1, 7
3610 NO = NO + 1
3620 RETURN
3630 LOCATE 20, 15: COLOR 1, 7: PRINT "Press "; : COLOR 10, 7:
       PRINT "a";
3640 COLOR 1, 7: PRINT ","; : COLOR 10, 7: PRINT " b";
3650 COLOR 1, 7: PRINT ","; : COLOR 10, 7: PRINT " c";
3660 COLOR 1, 7: PRINT " or"; : COLOR 10, 7: PRINT " d";
3670 COLOR 1, 7: PRINT " to answer    "
3680 RETURN
3690 DATA 1,2,3,4,5,6,7,8,9,10
```

```
3700 DATA 10,9,8,7,6,5,4,3,2,1
3710 DATA 5,4,6,3,7,10,1,2,9,8
3720 DATA 3,8,2,4,6,9,1,10,5,7
3730 DATA 7,3,6,5,1,8,4,9,10,2
3740 DATA 9,5,7,1,3,6,8,10,2,4
3750 DATA 4,1,3,6,2,10,2,7,5,9
3760 DATA 8,10,2,1,7,3,9,4,6,5
3770 DATA 2,7,5,10,9,4,6,8,1,3
3780 DATA 6,10,4,7,8,3,9,2,1,5
5001 COLOR 14, 0: LOCATE 25, 50: PRINT MIN;
5004 COLOR 13, 0: PRINT "minutes and";
5005 COLOR 14, 0: PRINT SEC;
5006 COLOR 13, 0: PRINT "seconds left ";
5007 LOCATE 1, 1: COLOR 1, 7
5008 SEC = SEC - 1: IF SEC < 0 THEN SEC = 59: IF SEC = 59 THEN
       MIN = MIN - 1
5009 IF MIN < 0 AND SEC < 0 THEN 670
5010 RETURN
5020 LPRINT " "
5030 LPRINT " "
5040 LPRINT SPC(34); "SCORE REPORT "
5050 LPRINT " "
5060 LPRINT SPC(7); "Name   : ";
5070 LPRINT NA$
5080 LPRINT SPC(7); "Id no. : ";
5090 LPRINT ID$
5100 LPRINT SPC(7); "Year   : ";
5110 LPRINT YE$
5120 LPRINT SPC(7); "Class  : ";
5130 LPRINT CL$
5140 LPRINT SPC(7); "Date of studying lessons : ";
5150 LPRINT DA$
5160 LPRINT SPC(7); "Starting time            : ";
5170 LPRINT TI$
5180 LPRINT SPC(7); "Date of test carried out : ";
5190 LPRINT DATE$
5200 LPRINT SPC(7); "Starting time            : ";
5210 LPRINT TI2$
5220 LPRINT SPC(7); "Time spent on lessons    : ";
5230 TI1$ = TI$
5240 H1 = VAL(LEFT$(TI1$, 2))
5250 M1 = VAL(MID$(TI1$, 4, 2))
5260 S1 = VAL(RIGHT$(TI1$, 2))
5270 H2 = VAL(LEFT$(TI2$, 2))
5280 M2 = VAL(MID$(TI2$, 4, 2))
5290 S2 = VAL(RIGHT$(TI2$, 2))
5300 IF H2 < H1 THEN H = (24 - H1) + H2 ELSE H = H2 - H1
5310 IF M2 < M1 THEN M = (60 - M1) + M2: H = H - 1 ELSE M = M2
       - M1
5320 IF S2 < S1 THEN S = (60 - S1) + S2: M = M - 1 ELSE S = S2
       - S1
5330 LPRINT H;
5340 LPRINT "hour(s)";
5350 LPRINT M;
```

```
5360 LPRINT "minute(s)";
5370 LPRINT S;
5380 LPRINT "second(s)"
5390 TI3$ = TIME$
5400 H3 = VAL(LEFT$(TI3$, 2))
5410 M3 = VAL(MID$(TI3$, 4, 2))
5420 S3 = VAL(RIGHT$(TI3$, 2))
5430 IF H3 < H2 THEN H = (24 - H2) + H3 ELSE H = H3 - H2
5440 IF M3 < M2 THEN M = (60 - M2) + M3: H = H - 1 ELSE M = M3
     - M2
5450 IF S3 < S2 THEN S = (60 - S2) + S3: M = M - 1 ELSE S = S3
     - S2
5460 LPRINT SPC(7); "Time spent on test       : ";
5470 LPRINT H;
5480 LPRINT "hour(s)";
5490 LPRINT M;
5500 LPRINT "minute(s)";
5510 LPRINT S;
5520 LPRINT "second(s)"
5530 LPRINT SPC(7); STRING$(80, 205)
5540 LPRINT SPC(7); "Item";
5550 LPRINT SPC(20); "Right";
5560 LPRINT SPC(20); "Wrong"
5570 LPRINT SPC(7); STRING$(80, 205)
5580 FOR I = 1 TO 10
5590 LPRINT SPC(8); I; TAB(33); RIGHT(I); TAB(58); WRONG(I)
5600 NEXT I
5610 LPRINT SPC(7); STRING$(80, 205)
5620 LPRINT SPC(29); "Sum";
5630 LPRINT R / 10;
5640 LPRINT SPC(19); "Sum";
5650 LPRINT W
5660 LPRINT " "
5670 LPRINT " "
5680  LPRINT  SPC(32);  "Average  ";  :  LPRINT  R;  :  LPRINT
     "percent"
5685 COLOR 7, 0: CLS
5690 END
```

LIST OF STORY SEQUENCE FOR "LIFEBOAT EQUIPMENT"

AUNGLIFE.SH

						SPRITE ACTION		COLOR	DIR	POSITION	
LABEL	COMMAND	PARAMETERS	TIME	WAIT	METHOD	DIR	LINE	AREA	FROM PICTURE		TO SCREEN
TITLE	/DISPLAY*	TUTITLE1.PIC-	0	0	INSTAN	NONE	NONE	FULL	11- 21	101-116	532-175
	/DISPLAY*	TUTITLE2.PIC-	FAST	0	WEAVE	NONE	NONE	TPART	9-394	264-456	9-394
	/DISPLAY*	TUTITLE2.PIC-	FAST	5	WEAVE	NONE	NONE	TPART	481-397	624-438	481-397
	/DISPLAY*	TUTITLE2.PIC-	FAST	10	WEAVE	NONE	NONE	TPART	281-401	472-458	281-401
INTRO	/DISPLAY*	INTRO .PIC-	0	0	INSTAN	NONE	NONE	FULL	11- 21	101-116	532-175
	/DISPLAY*	PROMPTXT.PIC-	FAST	0	REPLAC	LEFT	NONE	TPART	345-449	640-476	345-449
	/INPUT*	1	0	0							
	/EXEC*	REGISTER.EXE	0	0							
MENU	/DISPLAY*	MENU .PIC-	0	0	INSTAN	NONE	NONE	FULL	11- 21	101-116	532-175
INPUT	/INPUT*	1	0	0							
	/IF*0	END	0	0							
	/IF*1	GOSUB DESCR	0	0							
	/IF*2	GOSUB USE	0	0							
	/IF*3	GOSUB TEST	0	0							
	/GOTO*	MENU	0	0							
DESCR	/DISPLAY*	COMPASS1.PIC-	0	5	STRIPE	UP	NONE	FULL	11- 21	101-116	532-175
	/DISPLAY*	CONTITXT.PIC-	FAST	0	REPLAC	RIGHT	NONE	TFULL	1-445	639-479	532-175
	/DISPLAY*	PROMPTXT.PIC-	FAST	0	REPLAC	LEFT	NONE	TFULL	1-445	639-479	532-175
	/INPUT*	1	0	0							
	/IF*1	GOTO MENU	0	0							
COMP2	/DISPLAY*	COMPASS2.PIC-	0	10	PUSH	UP	NONE	FULL	1-445	639-479	532-175
	/DISPLAY*	PROMPTXT.PIC-	FAST	0	REPLAC	LEFT	NONE	TFULL	1-445	639-479	532-175
	/INPUT*	1	0	0							
	/IF*1	GOTO MENU	0	0							
	/*		0	0							
CRUTC	/DISPLAY*	CRUTCHTU.PIC-	0	0	EXPLOD	OUT-H	NONE	FULL	1-445	639-479	532-175
	/INPUT*	1	0	0							
	/IF*1	GOTO MENU	0	0							
	/*		0	0							
HANDF	/DISPLAY*	HANDFLA1.PIC-	0	0	FADE	NONE	NONE	FULL	1-445	639-479	532-175
	/INPUT*	1	0	0							
	/IF*1	GOTO MENU	0	0							
	/*		0	0							
	/DISPLAY*	HANDFLA2.PIC-	0	0	PUSH	LEFT	NONE	FULL	1-445	639-479	532-175
	/INPUT*	1	0	0							
	/IF*1	GOTO MENU	0	0							
	/*		0	0							
HATCH	/DISPLAY*	HATCHET .PIC-	0	0	SPLIT	OUT-H	NONE	FULL	1-445	639-479	532-175
	/INPUT*	1	0	0							
	/IF*1	GOTO MENU	0	0							
	/*		0	0							
HELIO	/DISPLAY*	HELIODES.PIC-	0	0	SPLIT	IN-V	NONE	FULL	49-195	615-421	49-195
	/DISPLAY*	OPHELI3 .PIC-	0	0	EXPLOD	IN-V	NONE	TPART	1- 1	392-212	1- 1
	/DISPLAY*	HELIOTXT.PIC-	0	0	PUSH	LEFT	NONE	TPART	41-259	599-353	41-259
	/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
	/INPUT*	1	0	0							
	/IF*1	GOTO MENU	0	0							
	/*		0	0							
JACK	/DISPLAY*	JACKNIFE.PIC-	0	3	FADE	NONE	NONE	FULL	1-445	639-479	532-175
	/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175

LIST OF STORY SEQUENCE FOR "LIFEBOAT EQUIPMENT"

AUNGLIFE.SH

						SPRITE ACTION			COLOR	DIR	POSITION	
	LABEL	COMMAND	PARAMETERS	TIME	WAIT	METHOD	DIR	LINE	AREA	FROM PICTURE		TO SCREEN
51		/INPUT*	1	0	0							
52		/IF*1	GOTO MENU	0	0							
53		/*		0	0							
54	OAR	/DISPLAY*	OAR .PIC-	0	0	DIAGON	UP	NONE	FULL	1-445	639-479	532-175
55		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
56		/INPUT*	1	0	0							
57		/IF*1	GOTO MENU	0	0							
58		/*		0	0							
59	LAMP	/DISPLAY*	LAMP2 .PIC-	0	0	EXPLOD	OUT-V	NONE	FULL	1-445	639-479	532-175
60		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
61		/INPUT*	1	0	0							
62		/IF*1	GOTO MENU	0	0							
63		/*		0	0							
64	PAINT	/DISPLAY*	PAINTERS.PIC-	0	0	CHECKE	NONE	NONE	FULL	1-445	639-479	532-175
65		/DISPLAY*	CONTITXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
66		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
67		/INPUT*	1	0	0							
68		/IF*1	GOTO MENU	0	0							
69		/DISPLAY*	PAINTE2A.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
70		/DISPLAY*	PAINTE2B.PIC-	MED	0	PUSH	LEFT	NONE	TPART	17-310	624-372	17-310
71		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
72		/INPUT*	1	0	0							
73		/IF*1	GOTO MENU	0	0							
74		/*		0	0							
75	PLUG	/DISPLAY*	PLUG .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
76		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
77		/INPUT*	1	0	0							
78		/IF*1	GOTO MENU	0	0							
79		/*		0	0							
80	PUMP	/DISPLAY*	PUMP2 .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
81		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
82		/INPUT*	1	0	0							
83		/IF*1	GOTO MENU	0	0							
84		/*		0	0							
85	RADIO	/DISPLAY*	RADIOCOL.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
86		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
87		/INPUT*	1	0	0							
88		/IF*1	GOTO MENU	0	0							
89		/*		0	0							
90	RATIO	/DISPLAY*	RATION .PIC-	0	0	DIAGON	LEFT	NONE	FULL	1-445	639-479	532-175
91		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
92		/INPUT*	1	0	0							
93		/IF*1	GOTO MENU	0	0							
94		/*		0	0							
95	ROCKE	/DISPLAY*	ROCKRTDE.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
96		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
97		/INPUT*	1	0	0							
98		/IF*1	GOTO MENU	0	0							
99		/*		0	0							
100		/DISPLAY*	ROCKEDE2.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175

LIST OF STORY SEQUENCE FOR "LIFEBOAT EQUIPMENT"

AUNGLIFE.SH

							SPRITE ACTION		COLOR	DIR	POSITION	
	LABEL	COMMAND	PARAMETERS	TIME	WAIT	METHOD	DIR	LINE	AREA	FROM PICTURE		TO SCREEN
1		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
2		/INPUT*	1	0	0							
3		/IF*1	GOTO MENU	0	0							
4		/*		0	0							
5	SEANC	/DISPLAY*	SEANCHO .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
6		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
7		/INPUT*	1	0	0							
8		/IF*1	GOTO MENU	0	0							
9		/*		0	0							
0	SMOKE	/DISPLAY*	SMOKECOL.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
1		/DISPLAY*	CONTITXT.PIC-	0	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
2		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
3		/INPUT*	1	0	0							
14		/IF*1	GOTO MENU	0	0							
15		/*		0	0							
16		/DISPLAY*	SMOKECO2.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
17		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
18		/INPUT*	1	0	0							
19		/IF*1	GOTO MENU	0	0							
20		/*		0	0							
21	THERM	/DISPLAY*	THERBLAN.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
22		/DISPLAY*	CONTITXT.PIC-	0	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
23		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
24		/INPUT*	1	0	0							
25		/IF*1	GOTO MENU	0	0							
26		/*		0	0							
27		/DISPLAY*	THERSUIT.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
128		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
129		/INPUT*	1	0	0							
130		/IF*1	GOTO MENU	0	0							
131	CARTO	/DISPLAY*	INTRO2 .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
132		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
133		/INPUT*	1	0	0							
134		/IF*1	GOTO MENU	0	0							
135	USE	/DISPLAY*	OPHELI1 .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
136		/DISPLAY*	CONTITXT.PIC-	0	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
137		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
138		/INPUT*	1	0	0							
139		/IF*1	GOTO MENU	0	0							
140		/*		0	0							
141		/DISPLAY*	OPHELI2 .PIC-	0	0	BLEND	NONE	NONE	FULL	1-445	639-479	532-175
142		/DISPLAY*	CONTITXT.PIC-	0	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
143		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
144		/INPUT*	1	0	0							
145		/IF*1	GOTO MENU	0	0							
146		/*		0	0							
147	1	/DISPLAY*	OPHELI3 .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
148		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
149		/INPUT*	1	0	0							
150		/IF*1	GOTO MENU	0	0							

LIST OF STORY SEQUENCE FOR "LIFEBOAT EQUIPMENT"

AUNGLIFE.SH

						SPRITE ACTION			COLOR	DIR	POSITION	
O.	LABEL	COMMAND	PARAMETERS	TIME	WAIT	METHOD	DIR	LINE	AREA	FROM PICTURE		TO SCREEN
51		/*		0	0							
52	2	/DISPLAY*	OPRADIO2.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
53		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
54		/INPUT*	1	0	0							
55		/IF*1	GOTO MENU	0	0							
56	3	/DISPLAY*	SEANCUSE.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
57		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TFULL	1-445	639-479	532-175
58		/INPUT*	1	0	0							
59		/IF*1	GOTO MENU	0	0							
60	4	/DISPLAY*	OPSEAPA2.PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
61		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TPART	1-445	639-479	1-445
62		/INPUT*	1	0	0							
63		/IF*1	GOTO MENU	0	0							
164		/*		0	0							
165	5	/DISPLAY*	OPSMOKE .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
166		/DISPLAY*	PROMPTXT.PIC-	FAST	0	CRUSH	OUT-H	NONE	TPART	1-445	639-479	1-445
167		/INPUT*	1	0	0							
168		/IF*1	GOTO MENU	0	0							
169		/*		0	0							
170	CLOSE	/DISPLAY*	CLOSE .PIC-	0	0	CRUSH	OUT-H	NONE	FULL	1-445	639-479	532-175
171		/INPUT*	1	0	0							
172		/IF*1	GOTO MENU	0	0							
173	TEST	/EXEC*	TEST .EXE	0	0							
174		/GOTO*	MENU	0	0							
175		/*		0	0							
176		/*Story Last Line		0	0							